



UNIVERSITA' DI PISA

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria informatica per la gestione d'azienda

TESI DI LAUREA

Il Knowledge management nel Lean PLM

RELATORI

Prof. Roberto CHIAVACCINI

Prof. Gigliola VAGLINI

CANDIDATO

Stefano ATZENI

Anno Accademico

2013/2014

Alla mia famiglia.

A Giulia.

Indice

| | |
|---|----|
| Abstract..... | 6 |
| Capitolo 1 | 7 |
| 1.1 Concurrent Engineering..... | 7 |
| 1.1.1 Serial Engineering..... | 8 |
| 1.1.2 Concurrent engineering..... | 11 |
| 1.2 Set-based concurrent engineering | 14 |
| 1.2.1 I principi..... | 16 |
| 1.2.2 Learning cycles | 23 |
| 1.2.3 Mitigazione del rischio..... | 24 |
| Capitolo 2 | 35 |
| 2.1 A3 | 35 |
| 2.1.1 Analisi della situazione corrente..... | 38 |
| 2.1.2 Identificazione delle Root Cause | 39 |
| 2.1.3 Identificazione delle Countermeasure..... | 40 |
| 2.1.4 Creazione del piano implementativo..... | 40 |
| 2.1.5 Lo strumento | 41 |
| Capitolo 3 | 48 |
| 3.1 Trade-Off Curves | 48 |
| 3.1.1 Come si costruiscono..... | 53 |
| 3.1.2 Costruire le trade-off curves..... | 60 |
| 3.1.3 Esempi pratici di utilizzo | 60 |
| Capitolo 4 | 67 |
| 4.1 Product lifecycle management | 67 |
| 4.1.1 Introduzione | 67 |
| 4.1.2 PLM | 67 |
| 4.1.3 Lifecycle..... | 76 |
| 4.1.4 I benefici nel business..... | 80 |
| 4.2 Aras | 83 |
| 4.2.1 ARAS Innovator in dettaglio..... | 87 |
| Capitolo 5 | 90 |
| 5.1 Implementazione in ARAS Innovator | 90 |
| 5.1.1 Introduzione | 90 |

| | |
|---|-----|
| 5.1.2 Modulo PLD (Product Lean Development) | 91 |
| 5.1.3 Implementazione delle metodologie..... | 93 |
| 5.1.4 Gli Item Types Utilizzati | 106 |
| 5.1.5 Relationship..... | 110 |
| 5.1.6 I metodi | 113 |
| Conclusioni..... | 118 |
| Ringraziamenti | 120 |
| Appendice | 121 |
| Lista delle figure | 121 |
| Bibliografia | 122 |
| Riferimento a libri..... | 122 |
| Riferimenti ad articoli..... | 122 |

Abstract

L'obiettivo del presente lavoro di tesi è l'implementazione di un modulo del software ARAS Innovator che permetta di supportare le metodologie note sotto il nome di Lean Product Development, che permettono di applicare le metodologie Lean, introdotte da Toyota, alla produzione manifatturiera.

Il software ARAS non mette a disposizione un modulo specifico per la gestione della progettazione e sviluppo in maniera "lean"; è stato quindi creato un nuovo modulo sfruttando sia le funzionalità già presenti nel software, sia creandone delle nuove per riuscire a raggiungere gli obiettivi della tesi.

Attraverso questo modulo sarà quindi possibile per un'azienda, creare e gestire un nuovo progetto, definire dei rischi associato a esso e gestirli attraverso i così detti "Learning cycle event". A ognuno di essi sarà associato uno specifico documento A3, sviluppato da zero all'interno del programma, rendendo possibile analizzare, gestire e confrontare le performance delle diverse soluzioni studiate per riuscire a mitigare ognuno dei rischi.

Lo studio partirà analizzando la metodologia di progettazione nota sotto il nome di set-based concurrent engineering, analizzando i benefici rispetto alle precedenti tecniche di progettazione industriale e come queste possono essere applicate; si studieranno e analizzeranno in dettaglio due diversi strumenti a supporto della progettazione: gli A3 e le trade-off curves. In seguito si farà una rapida introduzione ai software PLM, per poi passare all'analisi del funzionamento del PLM ARAS Innovator e alla descrizione del lavoro implementativo.

Tutte le tematiche analizzate nel presente lavoro di tesi si basano sugli studi effettuati da Ronald Mascitelli, e riportati sul libro "Mastering lean product development: A practical event-driven process for maximizing speed, profits and quality".

Capitolo 1

1.1 Concurrent Engineering

Nello scenario attuale le aziende devono essere in grado di adattarsi in maniera rapida ai cambiamenti del mercato, ai nuovi scenari competitivi e ridurre il *time to market*. Uno degli aspetti chiave è riuscire a prendere decisioni in maniera rapida ed essere sicuri che queste ultime siano corrette la prima volta che sono prese (se così non fosse, si correrebbe il rischio di non avere abbastanza tempo per pensare a un nuovo piano d'azione e prendere una nuova decisione).

Le aziende, quindi, non possono permettersi di prendere delle decisioni sbagliate e impiegare risorse ed energie in soluzioni o progetti che non portano a un risultato concreto: per questo motivo sempre più spesso si assiste all'introduzione nelle aziende del *concurrent engineering*.

Il *concurrent engineering* è tuttora considerato uno standard nella progettazione all'interno delle aziende, a prescindere dalla loro dimensione, dalla loro strutturazione o dal loro portfolio prodotti. È necessario che le aziende modifichino il loro comportamento e i loro processi per permettere di portare avanti diversi progetti in maniera che le varie funzioni lavorino in maniera concorrente.

Questa metodologia può essere vista come un approccio sistematico, integrato e concorrente alla progettazione (incluso anche i processi di supporto e di manifattura). Lo scopo ultimo di questo tipo di progettazione è di avere una visione a 360 gradi del progetto e dei problemi a esso correlati (si prendono quindi in considerazione diversi aspetti come qualità, costi, programmazione e requisiti del cliente).

1.1.1 Serial Engineering

Le vecchie metodologie di progettazione si basavano su un approccio che è stato chiamato *seriale*, in virtù del susseguirsi delle varie fasi in maniera sequenziale: in quest'approccio ogni fase del progetto inizia quando la precedente è stata conclusa. Durante l'esecuzione di questo processo, ogni dato della fase precedente (anche quelli incompleti) è passato alla fase successiva. Chiaramente il ciclo di vita o la durata dell'intera progettazione, rischia di allungarsi considerevolmente: la durata complessiva sarà un multiplo della somma delle durate delle diverse fasi.

Nella Serial Engineering non esiste una vera e propria progettazione consistente, non esistono analisi e metodologie documentate. Questo è dovuto al fatto che molto spesso la preparazione dei dati per condurre le analisi fa sprecare molto tempo, e può capitare che durante le attese per la raccolta dei dati vengano fatti dei cambiamenti al progetto che rendono l'analisi inutile e inefficace.

Oggi giorno è possibile utilizzare strumenti computerizzati quali CAD, CAM, CAPP e così via; tuttavia questo rende possibile automatizzare soltanto alcuni aspetti, ma le varie fasi di questo tipo di progettazione procedono comunque in maniera sequenziale. Ulteriore tempo è perso nella gestione delle diverse fasi e delle loro interfacce, portando alla creazione di "isole" all'interno dei dipartimenti, agevolati dalla tradizionale suddivisione delle aziende.

I punti a sfavore di questo tipo di progettazione possono così essere riassunti:

- Si basa sull'idea che nessuna fase può iniziare sin tanto che la precedente non è stata conclusa e approvata (allungando i tempi del ciclo di vita).

- La natura lineare della progettazione implica che i costi di produzione devono essere stimati in gran parte nelle fasi iniziali della progettazione, senza che questi dati siano supportati da dati oggettivi.
- A causa del tempo perso potrebbe accadere che il prodotto portato sul mercato non rispecchi più i requisiti del mercato.

1.1.1.1 Serial engineering: waterfall model

Il waterfall model è forse uno dei più famosi modelli lineari di progettazione. Come in tutti i modelli lineari, anche in questo, ogni fase deve essere completata per far partire la successiva. Questo tipo di progettazione è particolarmente utilizzato nell'ambito della progettazione software.

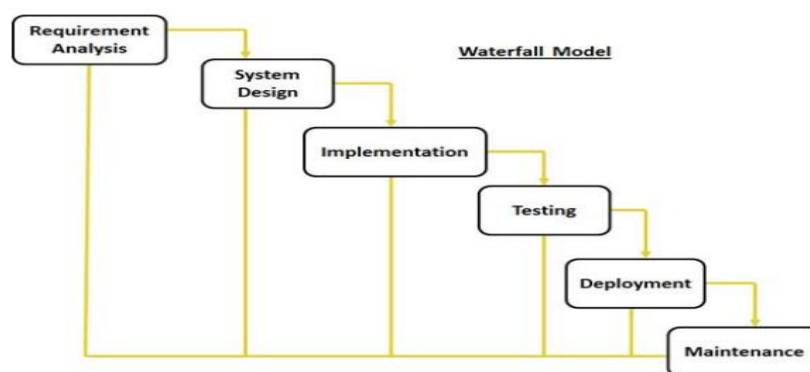


Figura 1.1: Waterfall model

La prima fase è quella che permetta la raccolta dei requisiti del sistema e la creazione di un documento che li contenga; vengono anche svolte le analisi dei requisiti. I requisiti sono quindi passati alla fase successiva in maniera tale che possano essere studiati e usati per redigere il progetto. Attraverso gli input della fase “System design” il sistema è sviluppato: all’inizio sono sviluppati piccoli programmi chiamati “units” che saranno integrati nella

fase successiva. Una volta svolta l'integrazione, il sistema è testato. La fase successiva è quella di provare il sistema all'interno dell'ambiente del cliente o rilasciarlo sul mercato. L'ultima fase è quella che permette di gestire e risolvere eventuali problemi che non erano stati trovati nelle fasi successive.

Come si può notare dalla schematizzazione di questo tipo di progettazione, esiste un arco che va da ciascuna fase alla precedente: questo sta a indicare che sono possibili rilavorazioni (che a seconda dei casi possono far allungare notevolmente i tempi di rilascio sul mercato del prodotto).

Il problema più grosso di questo tipo di progettazione, è che la fase di testing e di verifica del prodotto da parte del cliente finale, avviene solamente alla fine del processo: se non si è riusciti a soddisfare le aspettative del cliente, il processo incomincia nuovamente da capo.

I processi denominati "Waterfall" e "V" sono dei processi di sviluppo in cui prima si progetta il sistema, si trovano le interfacce tra i sottosistemi, dopo di che si progettano i sottosistemi. Funzionano top-down in fase di progettazione, e bottom-up in fase di testing. Questo tipo di progettazione, da un lato permette di progettare in maniera separata e indipendente i sottosistemi, dall'altro impone che le decisioni sulle interfacce siano fatte sulla base di dati "vecchi". Nella maggior parte dei progetti condotti secondo questo schema, si arriva a soluzioni "distorte".

Si può schematizzare il processo di sviluppo tradizionale come segue:

- Si definiscono le specifiche del progetto (o si ricevono dal cliente o dal mercato).
- Si producono una serie di concetti.
- Si sceglie uno di questi concetti.

- Il team aggiunge dettaglio al concetto prescelto, includendo le specifiche dei sottosistemi.
- Si testano i sottosistemi, e successivamente il sistema.
- L'intero processo è ripetuto per il sistema manifatturiero.

Scegliendo un concetto nella fase iniziale dello sviluppo si corre il rischio di prendere una decisione basata su dati vecchi, non permettendo la nascita di nuovi concetti o idee innovative.

1.1.2 Concurrent engineering

Per riuscire a risolvere i principali problemi della progettazione di tipo seriale è stata sviluppata, agli inizi degli anni '90, una nuova metodologia di progettazione. Questa si focalizza sull'analisi del tempo impiegato nel compiere le varie fasi della progettazione: lo scopo è di ridurre il tempo di progettazione senza perdere qualità. L'analisi è condotta andando a considerare tutte le aree coinvolte nella progettazione che aggiungono valore alla progettazione stessa. In questa metodologia le varie fasi procedono in parallelo, con cicli di feedback nel caso in cui ce ne sia bisogno.

Tradizionalmente lo sviluppo di nuovi progetti, o nuovi prodotti, nell'industria manifatturiera segue un rigido iter che prevede la focalizzazione su un'unica soluzione, generata per rispondere a uno specifico problema, o necessità. Questa modalità di lavoro può essere vista facendo riferimento a un insieme di funzioni che collaborano tra di loro, e aggiungono valore alla soluzione secondo il loro punto di vista.

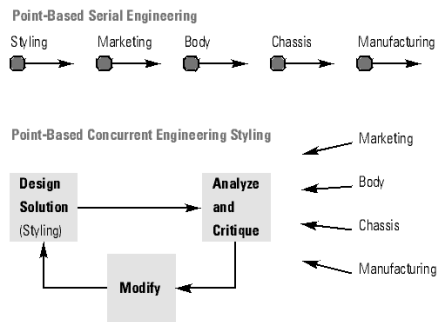


Figura 1.2: Tradizionale approccio Point-Based allo sviluppo prodotto

Come si può vedere dall'immagine si hanno una serie di funzioni: la funzione styling crea la propria soluzione secondo il suo punto di vista, dopo di che la funzione Marketing, secondo la soluzione proposta dalla funzione precedente, sviluppa il proprio progetto, e così via. Questo tipo di progettazione è noto in letteratura sotto il nome di “point-based serial engineering”. Nella realtà esistono numerosi cicli di feedback tra le soluzioni proposte dalle funzioni, ma questo non fa altro che aumentare il tempo e le rilavorazioni.

La seconda parte della figura sopra riportata, fa riferimento al tipico approccio Americano allo sviluppo: può essere visto come un raffinamento del “point-based serial engineering”. Secondo quest'approccio una funzione (in questo caso la funzione styling) genera una propria soluzione al problema e la mostra alle altre funzioni, che avranno il compito di analizzare la soluzione secondo il loro punto di vista. Mediante questa tecnica si ottiene un risultato migliore rispetto al precedente, perché tutte le modifiche possono essere fatte nella fase iniziale del progetto, senza costi aggiuntivi e perdite di tempo.

Quello che è chiaro in entrambi gli approcci è che la progettazione si focalizza soltanto su una soluzione: a un problema ne esistono di diverse, alcune peggiori di altre. Il problema chiaramente sorge nel momento in cui ci si focalizza nella soluzione sbagliata o peggiore: nella stragrande maggioranza dei casi ci si accorge dello sbaglio soltanto alla fine del processo.

Il team quindi potrebbe non riuscire mai a trovare la soluzione migliore: quello che può fare è verificare se la soluzione trovata è sufficientemente buona, ma questo soltanto alla fine del processo.

1.1.2 .1 Esempio di progettazione point-based: US Auto Body Development

La progettazione americana di automobili è il tipico esempio d'ingegneria point-based. Il processo manifatturiero e il processo di sviluppo sono composti di: sviluppo del concetto, prototipazione, strumenti di produzione e fase di produzione.

Nella prima fase, quella in cui si sviluppa il concetto del progetto, un piccolo gruppo di designers lavorano su un ristretto numero di progetti “grezzi” rappresentati diverse alternative per il progetto per il corpo macchina: alla fine del processo è scelto quella che secondo il loro punto di vista è la soluzione più idonea, la migliore. Dopo questa prima fase si costruisce l'architettura del veicolo secondo il progetto prescelto. La funzione manifatturiera è inglobata all'interno del processo di progettazione soltanto dopo che il progetto base è stato definito, anche se negli ultimi anni ci si accorti dell'importanza di avere le idee della funzione manifatturiera nelle prime fasi d'ideazione del progetto.

La fase successiva è quella che prevede di indicare possibili modifiche al progetto, utilizzando un modello di plastica. Nel momento in cui queste modifiche sono ritenute fattibili, s'incorporano all'interno del progetto, e si costruisce un secondo modello di plastica per ulteriori analisi. Dopo tre o quattro mesi questa fase si esaurisce, mettendo a disposizione del team l'architettura finale del veicolo. A questo punto sono possibili cambiamenti solamente nel caso in cui ci si accorga che certi pezzi non sono fabbricabili o

che non si rispettano gli standard di sicurezza minima. Anche l'ottimizzazione dei sottosistemi e dei componenti segue un processo simile.

Questo è un esempio di point-based design. Una singola soluzione è scelta nella fase iniziale della progettazione, dopo di che subisce delle modifiche a seconda delle esigenze, e del punto di vista, delle diverse funzioni aziendali.

1.2 Set-based concurrent engineering

Il set-based concurrent engineering adottato da Toyota differisce notevolmente dai precedenti aspetti perché si basa sulla scelta di una serie di possibili soluzioni (e non concentrando gli sforzi di progettazione su un'unica soluzione): lo scopo finale è quello di sfoltire questo set sino ad arrivare alla soluzione migliore.

Il processo è abbastanza semplice: i partecipanti discutono, sviluppano e comunicano una serie di soluzioni in maniera indipendente tra di loro. Conseguentemente si ridurrà questo set basandosi su una serie d'informazioni aggiuntive provenienti da studi, approfondimenti, test, richieste e opinioni dei clienti. Questa modalità di lavoro permette di far lavorare tutte le funzioni coinvolte in maniera perfettamente integrata, favorendo la comunicazione e limitando le rilavorazioni al minimo.

La robustezza di quest'approccio sta proprio nell'attenzione posta sulla comunicazione tra le diverse aree di un'organizzazione, e sulla flessibilità. Nello sviluppo di nuovi prodotti il fattore chiave è la flessibilità: questo è dovuto al fatto che il mercato negli ultimi anni è diventando sempre più imprevedibile e in continuo cambiamento. Il concetto di flessibilità si riferisce, in questa situazione, alla capacità di fare dei progetti modificabili in base alle

richieste del cliente (mercato) senza aggiunte di costo (o con aggravii di costo minimi) o perdite di tempo. Lavorare su diverse idee allo stesso momento favorisce la flessibilità.

Toyota applica questi concetti facendo lavorare i diversi team su diverse soluzioni e favorendo lo scambio d'informazioni in maniera esplicita. Lo scambio d'informazioni sui set di alternative e sulle regioni di fattibilità del progetto aumenta il valore aggiunto della comunicazione, favorendo al tempo stesso la diminuzione di lunghi meeting formali.

Per spiegare la filosofia che sta dietro al set-based concurrent engineering, si può far riferimento ai tre principi sotto elencati:

1. Map the design space
 - a. Define feasible regions
 - b. Explore trade-offs by designing multiple alternatives
 - c. Communicate sets of possibilities
2. Integrate by intersection
 - a. Look for intersections of feasible sets
 - b. Impose minimum constraint
 - c. Seek conceptual robustness
3. Establish feasibility before commitment
 - a. Narrow set gradually while increasing detail
 - b. Stay within sets once committed
 - c. Control by managing uncertainty at process gates

1.2.1 I principi

1.2.1.1 Map the design space

Con “Map the design space” s’intende il processo mediante il quale Toyota è solita sviluppare i set di alternative, che saranno poi gradualmente ridotti.

Le informazioni che è possibile ottenere da questo tipo di processo, scaturiscono dalla comunicazione di diverse alternative per un singolo progetto permettendo una maggiore comprensione delle possibilità, i margini di fattibilità e i vincoli di costo in maniera da riuscire ad avere una profonda conoscenza e comprensione del problema (in gergo l’output di questa fase è chiamato “design space”). In seguito si analizzeranno le diverse scelte andando a studiare i relativi documenti e le curve di trade-off.

Al contrario di quanto discusso in precedenza sul point-based engineering, ogni funzione coinvolta nel processo di sviluppo di un nuovo prodotto, in maniera relativamente indipendente e parallelamente, definisce delle regioni di fattibilità (quello che può essere o non può essere fatto o che non dovrebbe essere fatto) secondo il proprio punto di vista: il risultato è la creazione di diverse checklist. Le checklist non sono solo una serie di regole o di linee guida per il progetto: mediante esse si definiscono le attuali capacità così come sono state comprese dal responsabile del progetto. Queste sono usate per guidare la progettazione e per fare delle revisioni lungo le fasi del progetto stesso, per essere sicuri che il risultato rispetti certi requisiti minimi.

Questi passi permettono agli ingegneri la comunicazione di un set di alternative diverse, eliminando gli inconvenienti derivanti nel focalizzarsi solamente su un’unica soluzione. Il principio che sta alla base di questa metodologia è che una soluzione, che secondo un certo

punto di vista potrebbe essere vincente, potrebbe essere perdente secondo un altro punto di vista (secondo la visione o i requisiti imposti da un'altra funzione).

La creazione dei set di alternative aiuta la creazione di nuova conoscenza permettendo un'importantissima trasformazione delle conoscenze all'interno dell'azienda: trasformare le conoscenze tacite in conoscenze esplicite.

Attraverso questa metodologia si rende possibile la comunicazione delle alternative senza specificare valori esatti per i parametri del progetto, e in alcuni casi limite senza dover indicare neanche gli intervalli di confidenza. Una volta comunicati i set di alternative bisogna fare in modo di integrare queste diverse soluzioni tra di loro.

1.2.1.2 Integrate by intersection

Le funzioni, una volta comunicate le diverse possibilità, devono ricercare le intersezioni tra le stesse, per esempio andando a verificare dove le soluzioni trovate si sovrappongono. Se si riesce a trovare un'intersezione, si è trovata una soluzione che va bene per tutti. In Toyota prima di formalizzare un piano di azione si cerca il consenso tra le diverse parti (le funzioni) coinvolte: si creano dei meeting in cui si espongono le diverse soluzioni e si sollecitano i partecipanti nel discutere sulle soluzioni indicate, su quali sono i problemi o se può essere trovata qualche nuova soluzione, sino ad arrivare a un accordo tra tutte le parti.

Com'è stato detto in precedenza, Toyota è solita dare dei vincoli minimi al progetto. Tali vincoli sono molto flessibili, ed è possibile fare dei cambiamenti per permettere di aumentare l'integrazione. Quello che si vuole fare è prendere le decisioni al giusto

momento, e non all'inizio del progetto, come spesso è fatto nelle aziende manifatturiere occidentali.

Come si è detto è necessario che durante le prime fasi della progettazione s'individuino quali componenti e sotto sistemi compongono il sistema complessivo oggetto del progetto. Durante la ricerca bisognerà, però, tenere conto del fatto che molti componenti, in svariati sistemi, saranno molto integrati: il progetto dei sotto sistemi quindi condiziona il progetto complessivo e viceversa, e molte parti non saranno intercambiabili. Per riuscire a portare avanti questo tipo di progettazione è quindi indispensabile cercare di evitare il più possibile l'imposizione di vincoli, e imporli solamente nel caso in cui siano necessari.

Per spiegare meglio questo concetto si può fare un esempio. I body engineers di Toyota mandano i disegni CAD ai vari manufacturing engineers indicando solamente i valori nominali delle dimensioni, senza indicare la tolleranza.

Successivamente sono disegnati gli stampi permettendo di stampare le parti nella maniera più accurata possibile, facendo riferimento ai soli valori nominali indicati sui disegni. Mediante l'utilizzo di questi stampi sono stampate le parti, che saranno successivamente provate su di un veicolo vero e proprio. Il manufacturing team analizza le parti che sono state stampate per scoprire le imperfezioni e decidere quale stampo dovrà essere modificato per ottenere una parte che sia perfettamente collocabile sul prodotto finito. Successivamente si aggiorneranno i disegni CAD.

Lo stesso principio è utilizzato dalla casa automobilistica nipponica nei confronti dei propri fornitori: sono comunicati una serie di vincoli minimi; il fornitore avrà libera scelta di come affrontare la produzione del pezzo tenendo conto soltanto dei vincoli minimi.

Nelle realtà occidentali gli stampi sono fatti sulla base dei disegni CAD in cui sono indicati i valori precisi dei vari parametri. In questo caso i problemi risultano visibili soltanto nella

fase di assemblaggio (non si fanno prove, in quanto ci si fida dei valori indicati nei disegni), con un aggravio considerevole sui costi, e con una grossa perdita di tempo.

1.2.1.3 Establish feasibility before commitment

L'ultimo punto è lo studio della fattibilità della soluzione, prima che questa venga scelta. Come sottolineato, una volta costruito i set di alternative si deve cercare di “sfoltirlo” sino ad arrivare alla migliore soluzione possibile. Le funzioni riducono i loro set in parallelo, mantenendo aperta la comunicazione con le altre funzioni, in maniera tale da arrivare a una soluzione perfettamente integrata.

Il valore della comunicazione tra le funzioni riguardo i diversi set, si perde se uno o più partecipanti considera una soluzione che non è dentro il set che è stato comunicato. Quest'aspetto è particolarmente delicato, in quanto stando dentro i set, ogni membro del team è sicuro di poter procedere con il proprio lavoro senza preoccuparsi che questo possa generare rilavorazioni. Questo può essere raggiunto solamente se i set sono robusti, e cioè che al loro interno ci siano solamente soluzioni fattibili. Se la soluzione non è portata a termine entro una certa data, il team potrebbe ricorrere all'utilizzo di un'altra soluzione.

Toyota controlla il livello di fattibilità di ogni soluzione ai vari “Process Gates” (milestones).

È importante sottolineare che non esiste nessuna tecnica fondamentale per riuscire a implementare questa metodologia. I passi che sono stati appena descritti, non devono essere visti come dei passi prescrittivi: Toyota, infatti, applica tale metodologia personalizzandola sul prodotto o sul problema da risolvere.

È possibile schematizzare il set-based concurrent engineering nel modo seguente:

- Il team divide il sistema in sottosistemi, e i sottosistemi in parti ancora più piccole.
- S'identificano gli obiettivi generali per il sistema e i sottosistemi.
- Si creano diversi concetti, includendo aspetti sia del prodotto sia del sistema manifatturiero.
- Si filtrano i concetti trovati, cercando di capire i punti deboli. Si eliminano i concetti che non s'integrano con gli altri, con le aspettative dei clienti, la situazione competitiva, e così via.
- Le informazioni generate (comprese quelle che portano al fallimento del concetto) sono registrate e salvate sotto forma di trade-off curves.
- Mano a mano che il processo di sfoltimento va avanti, aumenterà l'accuratezza della soluzione, così che il concetto finale sarà affidabile e ottimizzato.

La set-based concurrent engineering si basa sui dati oggettivi permettendo di aumentare l'innovazione e la qualità del prodotto finale.

Molti sostengono che questa tecnica di progettazione non faccia risparmiare ne tempo ne denaro, ritenendo che aumentando il numero di soluzioni da analizzare faccia perdere tempo e il testing di ognuna di esse faccia aumentare drasticamente i costi. Questo a prima vista potrebbe essere vero.

Si deve però considerare, nella progettazione classica, il tempo che si perde nelle fasi di rilavorazione del progetto, quando arrivati in fondo ci si accorge che l'unica soluzione portata avanti non funziona. Oltre a questo si deve considerare che eseguire i test nella fase embrionale del progetto è incredibilmente più economico di quanto si possa pensare: per esempio Toyota costruisce un numero rilevante di prototipi nella fase iniziale, in maniera tale da capire quanto e se le diverse soluzioni sotto osservazione s'integrano tra di loro.

Solitamente sono necessarie soltanto una o due copie di questi prototipi; successivamente, nella fase di costruzione del prototipo principale, Toyota ha necessità di costruire soltanto 30-50 copie del prototipo, contro le centinaia delle compagnie tradizionali.

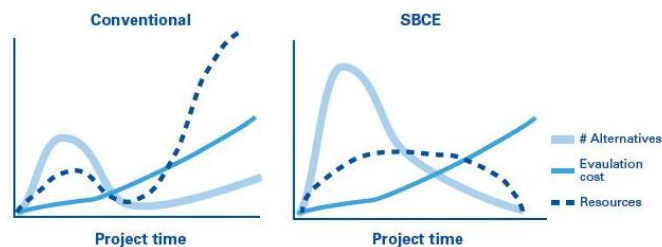


Figura 1.3: Traditional vs. Set-Based Concurrent Engineering

Un altro degli aspetti positivi è la “robustezza concettuale” delle soluzioni trovate. Una soluzione robusta, infatti, funziona bene in svariati sistemi e sono relativamente immuni alle variazioni nel resto del sistema: si scartano le soluzioni che funzionano soltanto in determinate condizioni o in determinati sistemi (per sistemi specializzati).

Considerando l'intero processo, la parte più delicata è l'eliminazione delle diverse alternative dal set. Per ognuno di questi concetti esisteranno delle cause che potrebbero farlo fallire o non funzionare a dovere. È quindi opportuno, per ognuno di esse, identificare queste cause e testare la soluzione in maniera tale da verificarne la robustezza.

Il testing può essere fatto in diverse maniere: svolgendo delle analisi, costruendo dei prototipi o facendo delle simulazioni. Chiaramente fare affidamento soltanto sulle cause, o sugli aspetti, che si pensano possano far fallire il concetto, non ha molto senso: se ne trovano dieci, ne esisterà sicuramente un'undicesima che farà fallire la soluzione e il progetto.

È quindi opportuno creare e costruire un prototipo per testare la soluzione nella maniera più veloce, economica e reale possibile.

Una delle cause di fallimento di un sistema costituito da diversi sottosistemi, e che questi ultimi non s'integrino tra di loro: sarà necessario fare dei test che ne provino l'effettivo funzionamento, e questi test dovrebbero essere fatti il prima possibile, e quindi nelle prime fasi della progettazione. Una delle possibili alternative quando si devono eseguire questi tipi di test, è l'identificazione di un insieme di clienti (dovrebbero essere identificati i clienti che concentrano nelle loro mani la fetta più grossa delle richieste, o che sono il target), e far testare a loro il prodotto in maniera da identificare le problematiche (per esempio gli autisti di taxi della Tunisia stressano maggiormente il proprio mezzo in una settimana rispetto a quanto facciano tutti gli altri autisti in un anno). È necessario quindi fare numerosi test, per trovare il punto in cui la soluzione fallisce, e per migliorare le conoscenze e i dati, che saranno utilizzati per migliorare le trade-off curves. Dopo di che saranno queste ultime a guidare il progetto.

L'utilizzo del set-based concurrent engineering porta con sé un diverso approccio all'identificazione e la gestione dei requisiti e delle specifiche. Nella progettazione convenzionale il team genera una lista di requisiti abbastanza dettagliata nella prima fase della progettazione. La progettazione è quindi indirizzata per riuscire a soddisfare i requisiti identificati; la generazione di nuove idee è vincolata dalla rispondenza a tali requisiti, così come l'identificazione dei test opportuni per verificare la soluzione.

Quello che deve essere evidenziato, anche in questo frangente, è la tempistica nella quale è fatta l'identificazione dei requisiti: essendo fatta nelle prime fasi del progetto, si avrà a disposizione una quantità d'informazioni abbastanza limitate. Risulterebbe molto più utile considerare la lista dei requisiti, non come una lista bloccata senza possibilità di modifica,

ma come una lista dinamica in grado di cambiare a seconda del grado di conoscenza del team: più si andrà avanti con il progetto più le informazioni cresceranno rendendo più probabile l'individuazione di nuovi requisiti o l'eliminazione di quelli non idonei.

La conoscenza, di cui si è parlato precedentemente, può essere generata in diverse maniere: aumentando la comprensione dei bisogni dei clienti, esplorando le diverse alternative per il progetto secondo un punto di vista multidisciplinare, eseguendo test e simulazioni e così via.

1.2.2 Learning cycles

Attraverso l'applicazione dei concetti della set-based concurrent engineering è possibile ottenere una più profonda conoscenza del problema che si deve affrontare. Il processo di sfoltimento delle soluzioni può essere dunque visto come un processo Darwiniano in cui si separa il forte (la soluzione migliore) dal debole (le soluzioni peggiori).

Questo processo porta alla definizione dei così detti “learning cycles”.

I learning cycles (cicli di apprendimento) possono essere visti come uno strumento per mitigare i rischi associati al progetto, chiudere i così detti “knowledge gaps” e accelerare il processo di sviluppo di nuovi progetti.

1.2.3 Mitigazione del rischio

L'attività di creare il piano di un progetto è una delle attività di maggiore utilità per un team a cui è assegnato un progetto: attraverso quest'attività il team ha la possibilità di controllare e di indirizzare il progetto verso la buona riuscita.

È sorprendente constatare come la maggior parte dei progetti che vengono seguiti nelle aziende non ha un preciso piano di sviluppo. Le ragioni di questa tendenza sono da ricercare nel fatto che molte aziende hanno adottato metodologie di sviluppo di progetti di tipo standard, nella speranza che questo basti per riuscire ad avere progetti e risultati di successo. Chiaramente seguire un metodo di sviluppo di tipo standard, quindi uguale per tutti i tipi di progetti, non è l'idea vincente.

Per riuscire a venire fuori da questa situazione si dovrebbe adottare un nuovo approccio alla pianificazione, prevedendo, sin dalle prime fasi del progetto, degli incontri tra i membri del team per stabilire un piano di esecuzione e di attenuazione dei rischi.

Tra i numerosi vantaggi di questo tipo di approccio è da segnalare che, non essendo una sola persona responsabile della pianificazione (nella maggior parte il team leader), tutti all'interno del gruppo sono a conoscenza delle scelte fatte in fase di pianificazione nonché dei vincoli necessari alla sua realizzazione, permettendo al tempo stesso di aumentare le probabilità di successo e di realizzare gli obiettivi del progetto.

Una delle cause che potrebbero far insorgere dei problemi, è quella legata alla diversa preparazione di un project management professional e i componenti del team (non è comunque da sottovalutare la capacità di apprendimento dei singoli membri del gruppo, agevolata anche dalla condivisione delle informazioni nella fase di pianificazione). Per

riuscire ad ovviare a questo tipo di problema, è necessario usare nuove tecniche di pianificazione per riuscire ad ottenere il Project Master Schedule: queste nuove tecniche sono tecniche visive.

Le tecniche adottate in questo nuovo approccio provengono dal value stream mapping. Molto spesso si perde del tempo per riuscire a far funzionare il value stream mapping e per riuscire ad adattarlo alla propria situazione aziendale. Molto spesso il fallimento è legato alla natura stessa del progetto: in un'azienda in cui sono presenti numerosi progetti che tra loro non hanno attività a comune l'uso di questa tecnica risulta completamente inefficace. Un'altra ragione che potrebbe far fallire l'applicazione del value stream mapping è che quando un team sviluppa la future-state value map per una linea di produzione fa delle ipotesi sulla capacità lavorativa che sarà impiegata. L'operatore sarà dunque formato in maniera tale che possa svolgere la propria mansione nel miglior tempo possibile.

Questa possibilità nello sviluppo di un nuovo prodotto non è presente: quando si pianifica il progetto di un nuovo prodotto, non si è a conoscenza delle attività e dell'ordine in cui queste devono essere eseguite, così come non si è a conoscenza della complessità e della scalabilità delle stesse.

Un possibile effetto negativo nell'utilizzo della nuova tecnica potrebbe essere la ripetizione del lavoro eseguito in altri momenti da un altro team o da altri team (o almeno per la parte che due, o più, progetti possono avere in comune).

I due sistemi possono essere adoperati all'interno della stessa azienda in maniera parallela. È molto probabile, infatti, che l'azienda riconosca tra diversi progetti l'esistenza di master schedule molto simili. In questa situazione, una volta identificati le categorie di progetti, si potrebbe usare un schedule template che mette insieme gli attributi a comune tra i diversi

progetti facenti parte della stessa categoria. Questo template sarebbe, quindi, disponibile per progetti futuri.

Riassumendo si può quindi dire che un'azienda, al cui interno sono gestiti e sviluppati progetti ripetitivi e che appartengono alla stessa categoria, dovrebbe investire nell'utilizzo del value stream mapping, mentre aziende che hanno a che fare con progetti diversi tra di loro, e che non possono essere riuniti in una o più categorie, dovrebbero usare la nuova tecnica di progettazione.

Le aziende quando hanno la necessità di sviluppare un nuovo prodotto hanno come obiettivo quello di produrre un bene che sarà acquistato da quanti più clienti possibili. Per riuscire a raggiungere quest'obiettivo la necessità primaria è quella di capire cosa i clienti vogliono e quali sono le loro esigenze. Quest'obiettivo sta diventando sempre più difficile da raggiungere perché negli ultimi anni si è assistito a una veloce e repentina modifica della richiesta del mercato, creando un mercato micro segmentato da differenti categorie di prodotto.

Il processo che porta alla commercializzazione di un nuovo prodotto parte dalla percezione di un'opportunità di mercato e finisce con la vendita. Questo è un processo molto complesso e critico per le aziende. I rischi in questo contesto sono definiti come "l'effetto dell'incertezza sull'obiettivo".

I rischi possono essere di diversa natura: ritardi del fornitore (che a loro volta causeranno ritardi di consegna e di entrata nel mercato); rilavorazioni causati da cambi; ri-allocazione di risorse e molti altri.

Fondamentalmente possono essere individuate quattro categorie di rischi:

1. *Market risks*: questo rischio è legato alla reale conoscenza delle richieste e delle necessità del cliente, e dunque del mercato, alle ipotesi di volumi di produzione fatte, dal prezzo scelto e così via.
2. *Technical Risks*: riguarda ogni problema di fattibilità che s'identifica, i progetti concorrenti, nuove tecnologie manifatturiere.
3. *Schedule Risks*: vincoli sulle risorse, lead time, disponibilità di attrezzature per la prototipazione.
4. *Cost/Quality Risks*: riguarda l'individuazione del giusto costo target in fase di produzione, e l'individuazione del processo più adatto per riuscire a raggiungere gli standard qualitativi e di costo imposti per quel prodotto.

A questi quattro rischi ne possono essere aggiunti altri, in base alle esigenze dell'azienda o del progetto.

Una definizione un po' più generica di rischio può essere quella che identifica come rischio qualunque cosa che può avere un impatto negativo o dannoso sul progetto.

Un progetto ben condotto dovrebbe, quindi, prevedere nelle prime fasi delle azioni di mitigazione del rischio: una delle prime azioni potrebbe essere quella di non commettere errori che sono stati commessi in progetti precedenti, e cioè apprendere dal passato.

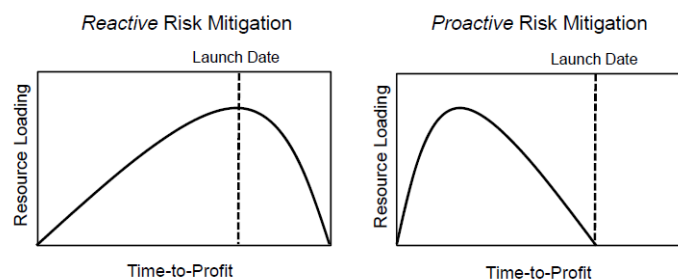


Figura 1.4: Differenze di approccio per la mitigazione dei rischi

La figura precedente mostra quest'approccio. Nel grafico di sinistra è mostrato un tipico progetto. Il team lavora in maniera normale sino a quando si verifica il primo problema, dopo di che il team entra nella fase di reazione aggiungendo risorse in maniera tale da cercare di riportare il progetto in carreggiata, ma durante questa attività potrebbero sorgere altri problemi. In questo contesto, anche sostenendo notevoli sforzi per riuscire a gestire i rischi, il progetto scivola oltre la data di consegna, facendo perdere fatturato e profitti.

Nel grafico di destra, invece, s'identificano preliminarmente i rischi e si affrontano quelli più critici, in maniera tale da risolverli. Tutti i rischi che potrebbero far sorgere un problema sono affrontati tramite i learning cycles. L'esecuzione dei cicli deve continuare sin tanto che il rischio persiste, o non è stato mitigato a un livello accettabile per il proseguo del progetto (il suo impatto sul progetto scende sotto la soglia massima).

I rischi hanno origine dall'incertezza presente in tutti i progetti e possono essere divisi in due grandi categorie: i rischi conosciuti e i rischi sconosciuti.

I rischi conosciuti, possono essere identificati, analizzati e risolti subito, in quanto si presuppone che, essendo già stati affrontati, esista un'efficace soluzione per mitigarli.

I rischi più difficili da identificare sono quelli che non si conoscono o che non sono mai stati affrontati nei precedenti progetti. Deve quindi essere creato un piano per riuscire ad affrontarli nella maniera più efficace ed efficiente possibile.

Le aziende possono essere disposte ad accettare diversi gradi per un rischio: questa è chiamata tolleranza al rischio. I rischi identificati all'interno di un progetto possono essere, quindi, accettati se ricadono all'interno del margine di tolleranza.

Il livello (o grado) del rischio considerato può essere visto come il prodotto di due fattori diversi: l'impatto che un determinato rischio può avere sul progetto, e la probabilità di

manifestazione del rischio sul progetto. Di solito la probabilità di occorrenza è maggiormente sotto controllo rispetto all’impatto. Infatti, possono essere prese contromisure per tenere sotto controllo questa proprietà: chiaramente attivando delle contromisure per cercare di abbassare la probabilità che il rischio si manifesti, non elimina il rischio stesso, ma si riduce la probabilità che il progetto vada alla derivata o che si rallenti il suo avanzamento.

Per la gestione di questi due parametri è possibile assegnare un valore a entrambi: per esempio, utilizzando la medesima scala, si potrebbero usare i valori che vanno da 1 a 5. Il valore 5 indicherebbe una “minaccia mortale” per il progetto, mentre un impatto con valore 1 indicherebbe un rischio facilmente assorbibile e gestibile rendendo il suo impatto minimo. Discorso simile è fatto per la probabilità di occorrenza, dove 5 indica una maggiore probabilità, mentre 1 indica la probabilità minima.

| | | |
|--|---|--|
| Impact (1-to-5 scale) – | | |
| 1 | - | Very minor risk – no significant project impact |
| 2 | - | Minor risk – can be managed without mitigation |
| 3 | - | Medium risk factor – may require mitigation |
| 4 | - | High risk factor – significant impact on cost / schedule |
| 5 | - | Very high risk factor – can be a “project killer” |
| Probability of Occurrence (1-to-5 scale) – | | |
| 1 | - | Very low probability – not worth considering |
| 2 | - | Low probability – very unlikely to occur |
| 3 | - | Medium probability – realistic chance of occurrence |
| 4 | - | High probability – likely to occur |
| 5 | - | Very high probability – almost certain to occur |

Figura 1.5: Il sistema 1-to-5

Una volta che sono stati identificati i rischi, e per ognuno di essi sono stati identificati i valori da associare alla probabilità e all’impatto, è possibile ordinarli in base al “Risk Priority Number” (RPN) ottenuto moltiplicando il valore associato al parametro “Impact” e il valore associato al parametro “Probability of Occurrence”. La figura seguente mostra un esempio di lista prioritizzata di rischi.

| Potential Risks to Project | Subjective Score | | Ranking (P x I) |
|--|------------------|---|--------------------|
| | P | I | |
| Retailers reject our electric bicycle and refuse to display them | 3 | 5 | 15 |
| Our price-point should be lower due to import competition | 4 | 2 | 8 |
| Proportional assist power system – feasibility | 5 | 5 | 25 |
| Miniaturizing control electronics to fit bicycle form factor | 2 | 3 | 6 |
| Lead time for UL submission and possible design corrections | 3 | 3 | 9 |
| Availability of critical environmental-testing lab equipment | 2 | 4 | 8 |
| Cost of tooling for injection-molded power-system enclosure | 4 | 2 | 8 |
| Supplier capability to provide source inspection of motor assy. | 3 | 2 | 6 |
| Too many features – added complexity will reduce reliability | 2 | 4 | 8 |
| Bicycle frame not ridged enough to support power assist | 1 | 5 | 5 |
| Product will be perceived as being for “sissies” | 2 | 4 | 8 |

Figura 1.6: Un esempio di rischi identificati per il progetto di una bici elettrica

Questa semplice tabella è perfettamente in linea con il principio dei manager Toyota di favorire gli strumenti visuali piuttosto che report o tabelle scritte; la figura successiva mostra invece com'è possibile sostituire i valori numerici per il risk priority number, con una figura circolare di diversi colori (estremizzando il concetto visual management).

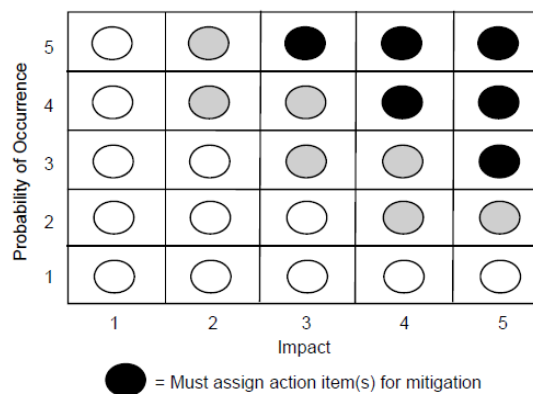


Figura 1.7: Rappresentazione grafica del sistema 1-to-5

Attraverso la lettura di questa tabella è possibile capire che i rischi che hanno un punteggio molto alto sia per la probabilità sia per l'impatto devono essere analizzati il prima possibile. I punti colorati in scuro (e cioè aventi un valore numerico elevato) dovrebbero essere affrontati immediatamente utilizzando degli strumenti di mitigazione del rischio.

Chiaramente i rischi possono essere affrontati in diverse maniere: il concetto fondamentale è però che possono essere mitigati o abbassando la probabilità o abbassando il valore associato al parametro dell'impatto.

L'identificazione dei rischi è un processo che può includere diverse figure professionali all'interno dell'azienda: project manager, project team members, risk management team, clienti, esperti, e anche componenti esterni all'azienda. L'identificazione del rischio non ricade solamente su queste figure professionali ma tutte le persone coinvolte nel progetto devono essere incoraggiate nell'identificazione dei rischi. L'identificazione può essere dunque vista come un processo iterativo, perché nuovi rischi possono essere identificati durante il ciclo di vita del progetto.

I rischi che dovrebbero raccogliere le maggiori attenzioni del team sono quelli che non sono stati perfettamente capiti, e per quali non esiste ancora un piano per poterli mitigare. In particolar modo si deve prestare grande attenzione all'identificazione delle cause radici del rischio: se queste non sono attentamente identificate, si possono creare i così detti *“knowledge gaps”*.

Questo tipo di rischio ha una grande probabilità di manifestarsi anche in progetti futuri, e quindi è giustificato il fatto di applicare un processo sistematico per riuscire a risolvere il problema e, al tempo stesso, costruire una solida base di conoscenza per i progetti futuri.

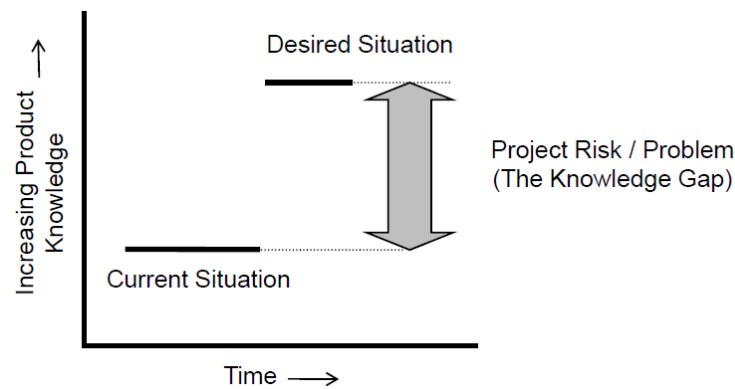


Figura 1.8: Un gap d'informazione esiste quando c'è disparità tra l'informazione corrente di un team e le informazioni necessarie per ottenere un progetto con bassi rischi

Una volta che sono stati identificati i rischi per cui è necessario agire subito, e sono stati individuati i Knowledge gap associati, è necessario studiare e implementare un piano di azione per la mitigazione del rischio: tale piano di azione prende il nome di “learning cycles”.

Il *learning cycle* è sostanzialmente un mini progetto al quale è associato un team. Il team avrà il compito di stabilire un piano per apprendere, definire un insieme di possibili soluzioni e stabilire una data entro la quale integrare le nuove conoscenze e valutare l'efficacia della soluzione implementata. È importante prestare particolare attenzione alla durata del learning cycle: questa dovrebbe essere decisa basandosi sulla tipologia di prodotto o progetto e sulla quantità di conoscenza che è necessario generare per arrivare alla soluzione desiderata.

La durata tipica di questi progetti è di circa un mese, anche se è se sarebbe preferibile riuscire a completarli in minor tempo.

Com'è stato detto precedentemente lo scopo è di acquisire nuova conoscenza all'interno dell'azienda (e del team in particolare) per riuscire a chiudere, o ridurre, il gap, in maniera tale da riuscire a gestire il rischio. Alla fine di ogni ciclo si andrà a controllare se il team è

riuscito nell'intento di chiudere il gap d'informazioni: se sono state acquisite un numero d'informazioni tali per cui è possibile raggiungere quest'obiettivo, il team può passare alla progettazione del nuovo prodotto, supportato dalle informazioni acquisite e da un'alta probabilità di riuscire ad ottenere una soluzione adeguata al primo tentativo. Nel caso in cui le informazioni acquisite non siano sufficienti a colmare il gap sarà necessario attivare un nuovo ciclo di apprendimento.

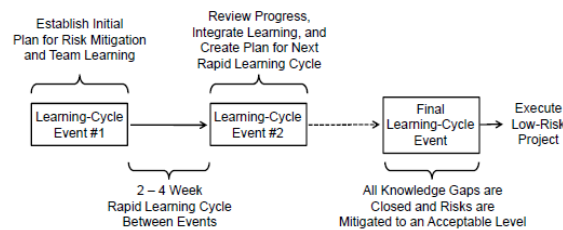


Figura 1.9: Il numero di iterazione dipende dalla complessità del problema che si sta affrontando

Questo tipo di gestione dei rischi può portare all'esecuzione di diversi cicli: il numero varia secondo la complessità del progetto. Per progetti relativamente semplici tutti i gap possono essere chiusi attivando uno o due cicli; per progettazioni di sistemi complessi potrebbe essere necessario attivare diversi cicli di apprendimento a livello di sistema, per riuscire ad ottenere un sistema ottimale, per poi procedere con l'attivazione di ulteriori cicli per i sottosistemi e per i componenti.

Da quest'ultima analisi si può capire come l'attivazione dei learning cycle non si esaurisce nella prima fase di analisi e gestione dei rischi, ma può continuare anche per le fasi successive della progettazione.

Attraverso questa tecnica il team impegnato nella progettazione ha la possibilità di acquisire una quantità notevole d'informazioni: questo vantaggio potrebbe andare completamente perso nel momento in cui il team è sciolto e le informazioni non sono salvate in una forma tale per cui tutti all'interno dell'azienda possano usufruirne.

È necessario codificare queste informazioni e renderle disponibile per progettazioni future, in maniera tale che un rischio che è stato mitigato per un progetto precedente, non debba essere nuovamente studiato per un progetto successivo, permettendo di accorciare il tempo necessario per progettazione.

Nella figura seguente si mostra un tipico esempio di agenda per un learning cycle: oltre ad identificare gli obiettivi e gli output, s'identificano i membri del team che parteciperanno al ciclo e l'agenda vera e propria.

| <u>The Learning-Cycle Event</u> | |
|---|---|
| Objective: To focus the collective knowledge of a cross-functional team on the mitigation of critical project risks and the elimination of knowledge gaps prior to executing a new product design. | |
| Inputs: | <ul style="list-style-type: none">• Market Requirements Brief (updated)• Engineering requirements matrix• Prioritized list of project risks |
| Outputs: | <ul style="list-style-type: none">• Risk-mitigation action list for critical project risks• Knowledge briefs documenting tradeoffs and solutions to knowledge gaps• Archived learnings of the team, including performance, curves, concept selection matrices, etc. |
| Attendees (typical): | <ul style="list-style-type: none">• Team leader• Team members (cross-functional development team)• Others (strategic suppliers, supply chain, quality, etc.) |
| Agenda (typical): | |
| 8:00 – 9:00 | Review prioritized risks and select critical issues |
| 9:00 – 11:00 | Sorting of risks into "immediately actionable" and "knowledge gaps" |
| 11:00 – 12:00 | Create / revise plan for development of knowledge briefs |
| 12:00 – 4:00 | Collaboratively discuss knowledge briefs and assign responsibilities and dates for next learning cycle |
| 4:00 – 4:30 | Identify actions to close out event |
| 4:30 – 5:00 | Management outbriefing / learning opportunities |

Figura 1.10: Un tipico documento che sintetizza l'agenda di un learning cycle

Capitolo 2

2.1 A3

La creazione dei cicli per la mitigazione dei rischi è giustificata dall'importanza del riuscire a scoprire le cause alla base della generazione del rischio, sia per il progetto attuale, sia per i progetti futuri. Non solo. La creazione e l'attivazione di questi tipi di progetto porteranno alla creazione di una quantità d'informazioni notevoli, che permetteranno all'azienda di aumentare le competenze e le conoscenze al proprio interno.

Uno degli strumenti di maggiore diffusione per la gestione di questo tipo di attività è l'A3. Il nome dello strumento deriva dalla dimensione del foglio utilizzato (anche se oggi giorni si è soliti utilizzare il più comune formato A4).

Il suo utilizzo è abbastanza semplice: al suo interno si troveranno informazioni sul problema che s'intende affrontare, un'analisi per identificare le cause che stanno alla base della sua manifestazione, e la documentazione della soluzione in maniera chiara, spesso accompagnata da illustrazioni o grafici. Lo scopo finale è quello di riuscire a creare in un unico foglio di carta, una storia che racconti qual è il problema, quali sono le cause e quali sono le soluzioni. Il fatto che tutte le informazioni siano in unico foglio di carta non è da sottovalutare: le stesse informazioni potrebbero essere raccolte in un report di diverse pagine, ma non avrebbero la stessa facilità di comprensione e l'immediatezza che si riesce a raggiungere attraverso l'utilizzo dello strumento A3 (anche questo concetto è in linea con il concetto di visual management di Toyota).

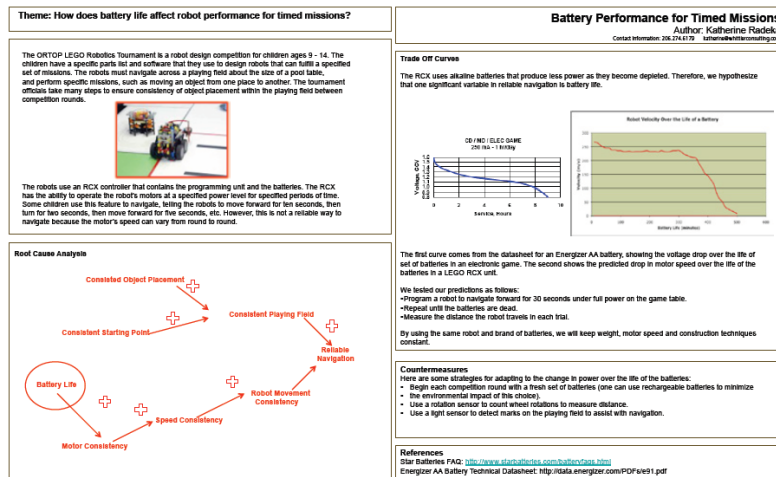


Figura 2.1: Un tipico documento A3

Solitamente in ogni A3 sono riportati il responsabile del progetto e la revisione del documento. La gestione della revisione del documento assume particolare importanza, in quanto, essendo un processo iterativo, sorgerà la necessità di continue modifiche prima di raggiungere una soluzione permanente. Le rimanenti sezioni del documento vanno dell'identificazione del problema, sino a un piano per riuscire a implementare la soluzione trovata.

Toyota utilizza questo strumento per implementare lo Shewart Cycle per il miglioramento continuo: noto anche come ciclo PDCA (Plan-Do-Check-Act), è un metodo in quattro steps per la gestione e il miglioramento continuo di processi e prodotti.

Nella prima fase, *PLAN*, s'identifica un piano di azione per riuscire ad affrontare un problema; nella fase *Do* s'implementa il piano trovato al passo precedente, collezionando dati e grafici; nella fase *CHECK* si fa uso dei dati e dei grafici trovati nella fase precedente, confrontandoli con i risultati attesi: in questa fase si possono identificare delle modifiche al piano per riuscire a raggiungere i risultati attesi; nell'ultima fase, *ACT*, s'implementano delle azioni per correggere il comportamento del piano, permettendo di raggiungere gli obiettivi prefissati.

Il paragone tra A3e PDCA può essere fatto andando a osservare come le prime due sezioni dell'A3 (identificazione del problema e delle cause) possono essere visti come la fase PLAN. La fase Do consiste nell'identificazione dell'obiettivo, l'identificazione di diverse alternative e la selezione delle contromisure; la fase CHECK è affrontata nella sezione "Verification Methods" dell'A3; l'ultima fase, quella di ACT, è la sezione "Implementation and follow-up plan".

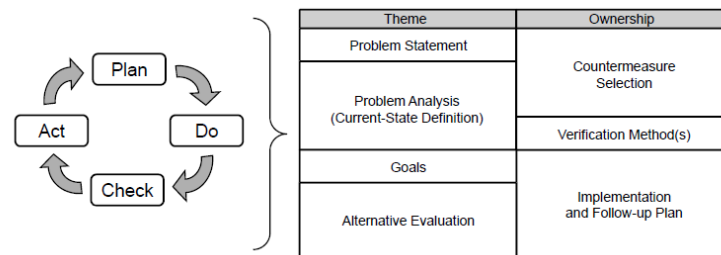


Figura 2.11: L'A3 è un metodo strutturato per l'implementazione del ciclo Plan / Do / Check / Act, per il miglioramento continuo

Com'è stato detto precedentemente, Toyota utilizza il report A3 per implementare la metodologia del miglioramento continuo: esiste un'altra fondamentale motivazione che ha spinto Toyota nell'utilizzo di questo strumento, ossia lo sviluppo intellettuale dei propri dipendenti. Infatti, è molto importante che i lavoratori Toyota attuino il pensiero razionale sia nel processo decisionale, sia nel problem solving.

La struttura e le tecniche che stanno alla base della costruzione dell'A3, favoriscono questo tipo di comportamento. Un altro aspetto da considerare è la capacità, attraverso questo strumento, di gestire i diversi punti di vista che si possono generare quando si affronta un problema: per esempio due diverse funzioni potrebbero vedere un problema solo secondo la loro prospettiva, che è per natura altamente soggettiva. Usando l'A3 si ha uno strumento esplicito, che rende possibile condividere il proprio punto di vista e di dimostrarlo. L'A3 è, dunque, uno strumento oggettivo.

Una delle caratteristiche principali del pensiero dei managers Toyota è la ricerca costante del consenso all'interno dell'azienda. Con il consenso si rende possibile eseguire il proprio lavoro, essendo sicuri che nessuno lo ostacoli e che tutti lavorino verso lo stesso obiettivo. L'A3 rende possibile, attraverso l'identificazione concreta del problema, delle cause, e delle possibili soluzioni, la comunicazione tra le parti, in maniera tale da arrivare a una soluzione allineata tra tutti. L'allineamento spesso coinvolge tre diverse comunicazioni: orizzontale tra le diverse funzioni, su e giù attraverso la gerarchia organizzativa e avanti e indietro nel tempo. La comunicazione orizzontale permette di incorporare le indicazioni tra i diversi componenti del gruppo; la comunicazione verticale permette di ottenere importanti informazioni dalla linea produttiva e dal top management; la comunicazione "temporale" si riferisce al fatto che è importante prendere in considerazione tutta la storia del problema.

Successivamente si dà una descrizione per ognuna delle sezioni dell'A3.

2.1.1 Analisi della situazione corrente

Il primo passo da compiere è quello di individuare e capire il processo che fa insorgere il problema, e individuare il suo contesto. In Toyota solitamente si procede andando ad analizzare fisicamente dove si presenta il problema, intervistando le persone coinvolte e capendo sotto quali condizioni si genera. Una volta che il team ha sufficienti informazioni sul problema, è possibile generare un disegno o un grafico della situazione corrente (nel caso in cui il problema non possa essere riportato sotto forma di disegno, si dovrà procedere alla descrizione in forma scritta). Fatto ciò sarà possibile condividere queste

informazioni con gli altri in maniera tale da essere sicuri dell'accuratezza delle ricerche effettuate e della comprensione del problema, ed eventualmente modificarle.

2.1.2 Identificazione delle Root Cause

In molti casi può essere immediato trovare delle cause responsabili di far generare il problema. È ancora abbastanza comune che queste cause non siano quelle fondamentali. Chiaramente per riuscire a identificare una soluzione adatta a risolvere il problema bisogna identificare i fattori chiave che lo generano, in maniera tale da eliminarli e prevenire in tal modo future manifestazioni del problema.

Uno dei più famosi metodi per l'identificazione dei fattori chiave (root cause) è il metodo delle *five whys*. Il metodo consiste nel porsi cinque volte la domanda perché, in maniera tale da arrivare alla radice del problema e identificarne le cause. Alla fine il problem solver (o il team) ha una profonda conoscenza della catena delle cause e degli effetti che si legano al problema. Anche in questo caso si dovrà procedere alla condivisione di queste informazioni per essere sicuri che le informazioni trovate siano probabili e ragionevoli.

È da notare come la tecnica dei *cinque perché* sia costretto da un importantissimo vincolo: da per scontato che ogni volta che viene data risposta alla domanda ci sia un'unica possibile risposta. Questo non è sempre vero, poiché nella vita reale i problemi sono una manifestazione di fattori multipli. Questo potrebbe far generare risposte diverse secondo la persona, impedendo di trovare un'unica risposta alla domanda perché.

2.1.3 Identificazione delle Countermeasure

Una volta che sono state identificate le root causes è necessario incominciare a pensare come queste possono essere affrontate, per impedire che insorgano problemi. Il nome “countermeasure” è stato introdotto da Toyota per indicare le azioni necessarie per impedire la comparsa, o ricomparsa, del problema.

2.1.4 Creazione del piano implementativo

Uno degli aspetti più importanti quando si vuole essere sicuri che un progetto sia vincente è il piano implementativo. Il piano può essere visto come una lista di azioni necessarie per rendere reale quello che sino a questo punto è rimasto su un pezzo di carta, individuando chi è il responsabile, la data d’inizio di ogni attività e la durata delle stesse. Il piano dovrà essere redatto dal responsabile del progetto insieme al personale coinvolto nel piano in maniera tale da raggiungere un accordo su azioni e tempi.

Una volta che il piano è stato approvato, si passerà all’esecuzione vera e propria. Durante l’esecuzione il team controlla se i risultati che ottiene collimano con quelli che erano previsti nel piano.

2.1.5 Lo strumento

Come detto precedentemente lo strumento A3 è un semplice foglio strutturato, utilizzato per implementare la metodologia PDCA e condividere le informazioni con tutti i dipartimenti all'interno dell'azienda. L'importanza di utilizzare questo tipo di strumento è oggi più importante di quanto non lo sia mai stato: le informazioni possedute dalle aziende crescono di giorno in giorno, avere uno strumento che raggruppa in un unico foglio una grande quantità d'informazioni diventa fondamentale.

Secondo il pensiero di Isao Kato, manager di Toyota, l'utilizzo dello strumento è stato pesantemente influenzato da diversi fattori. Uno dei maggiori fattori è stata l'introduzione delle metodologie PDCA e TQC (Total Quality Control). Quest'ultimo metodo ha introdotto metodi statistici rigorosi per il controllo statistico della qualità. Un altro fattore fondamentale è la marcata propensione del management Toyota per gli strumenti visuali a scapito di lunghi report scritti.

Il vantaggio competitivo nell'usare questo tipo di strumento e la facilità di condivisione delle informazioni: per ottenere il massimo risultato è necessario che venga supportato da una rigorosa politica aziendale di condivisione delle informazioni e di volontà al costante apprendimento. Lo scopo finale non è quello di implementare un nuovo strumento di gestione che permetta all'azienda di migliorare le proprie performance, ma di sviluppare un sistema flessibile che in maniera simultanea migliori le competenze della persona e le sue conoscenze, affrontando i problemi che ostacolano il cammino dell'organizzazione verso il miglioramento continuo.

A differenza di quanto si potrebbe pensare, non ne esiste un processo standard che porta alla creazione di un documento A3. Studi condotti tra i manager di Toyota, hanno

dimostrato che, per ogni specifico problema o progetto, viene utilizzata una versione personalizzata dell'A3. Lo scopo finale è quello di usare questo strumento nel lavoro giornaliero per riuscire a presentare e proporre delle soluzioni: non si vuole costruire degli A3 precisamente aderenti a uno specifico standard. L'unico standard cui fare riferimento è il modello PDCA.

2.1.5.1 Theme

Ogni A3 inizia con un titolo (un tema) che introduce l'argomento trattato. Questo dovrebbe permettere all'utilizzatore di capire l'obiettivo e il tema trattato all'interno dell'A3. Se lo strumento è utilizzato in forma elettronica, il tema diventa il campo su cui fare le ricerche. Uno degli errori da non commettere è vedere il problema sotto la prospettiva di una particolare soluzione: il tema dovrebbe focalizzarsi sul problema in generale, evitando di fare riferimenti a una particolare soluzione.

2.1.5.2 Problem Statement

Una volta che è stato definito un tema, l'autore deve documentare ogni informazione di background che è essenziale per comprendere l'importanza del problema. Se le informazioni inserite in questa sezione non sono chiare o non sono complete si corre il rischio che chi andrà a leggere il documento non capisca qual è il problema e debba richiedere informazioni aggiuntive, dando vita a una perdita di tempo. Un altro punto fondamentale è che l'obiettivo del report deve essere in linea con quelli che sono gli

obiettivi generali dell'azienda: ci si deve concentrare solo su gli obiettivi e i problemi più importanti per l'azienda (altrimenti si genera altra perdita di tempo). Altre informazioni che potrebbero essere utili in questa sezione riguardano i motivi che hanno reso possibile accorgersi del problema, le diverse parti coinvolte, i sintomi del problema, le performance passate, e così via.

2.1.5.3 Problem Analysis

Lo scopo di questa sezione è di inquadrare in maniera precisa e sintetica la situazione attuale. In questa sezione è facile trovare una rappresentazione visuale (grafici, tabelle, disegni, ecc.) che mostrano gli elementi critici del sistema o del processo che danno vita al problema. Lo scopo è di permettere al lettore di farsi un'idea precisa della situazione attuale, senza dare nessuna indicazione su possibili soluzioni. Avere una figura che mostra la situazione corrente facilita di gran lunga l'allineamento tra le parti coinvolte.

Dovrebbe essere messa particolare attenzione alla raccolta dei dati storici usati per la costruzione della situazione corrente, anche se resta essenziale riuscire a collezionare dei dati attuali per riuscire a essere il più dettagliato possibile. La costruzione di diagrammi facilita la lettura e la comprensione della situazione: l'autore è facilitato nella costruzione della sezione in quanto ha a disposizione un diagramma per mettere assieme un numero considerevole d'informazioni e dati; la lettura del diagramma consentirà ai lettori di capire in maniera immediata qual è il problema principale; la costruzione dei diagrammi permette di focalizzarsi sul sistema, consentendo un approccio maggiormente oggettivo. In molte situazioni lo sforzo per riuscire a risolvere un problema fallisce in quanto il team non riesce a capire qual è la situazione attuale.

2.1.5.4 Goals

La sezione “Goals” permette di rispondere a due obiettivi fondamentali: riuscire a capire che il progetto ha raggiunto gli obiettivi e quali standard sono stati utilizzati per fare le comparazioni. Questa sezione è di vitale importanza, quindi, per riuscire a capire se lo sforzo erogato per riuscire a risolvere un problema, è stato utile oppure meno (in buona sostanza se gli obiettivi sono stati raggiunti oppure meno).

Esistono dei punti da tenere in considerazione se si vuole costruire un efficace sezione “Goals”:

- Indicare degli obiettivi il più chiaramente possibile.
- Indicare delle chiare misure per misurare le performance.
- Considerare come collezionare i dati per valutazioni future, e controllare che le azioni intraprese si efficienti.

2.1.5.5 Root Causes

Uno degli aspetti di maggiore importanza nella costruzione dell’A3 (o comunque nella comprensione e nella ricerca di una possibile soluzione del problema) è l’identificazione delle cause che generano il problema. Non riuscire a identificare chiaramente, e in dettaglio, queste cause, sicuramente porterà a nuove manifestazioni del problema, vanificando in questo modo lo sforzo erogato per cercare di risolvere il problema. Come precedentemente indicato uno degli strumenti più comunemente utilizzati, è quello delle “5

Whys”. Considerando i problemi indicati in precedenza, è possibile utilizzare una tecnica alternativa per riuscire a rintracciare le cause radici del problema: il diagramma a lisca di pesce. Il diagramma, così come si vede nella figura seguente, è costruito con una serie di lische, ognuna rappresentante una possibile categoria di causa.

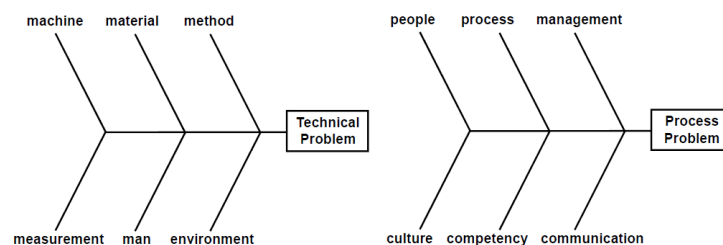


Figura 2.12: Il classico diagramma a spina di pesce, per la decomposizione di un problema nelle sue potenziali cause radici

Ogni persona coinvolta nella risoluzione del problema potrebbe investigare una categoria: una volta che sono state identificate una serie di possibili cause, è possibile ottenere una robusta base di partenza per l'analisi del problema. La causa, o le cause, ritenute maggiormente importanti possono essere selezionate per ulteriori approfondimenti.

Una possibile variazione a questo metodo potrebbe essere quello di inserire una probabilità per ogni causa trovata. In questo contesto per ognuna delle lische nel diagramma viene identificata una probabilità.

2.1.5.6 Countermeasure

Una volta che il team ha una profonda conoscenza della situazione attuale e delle cause che generano il problema, si può passare alla definizione delle possibili soluzioni, quelle che in Toyota chiamano countermeasures. La sezione sarà composta da una lista di possibili

soluzioni al problema, descrivendole in maniera tale da identificare le cause del problema, come sono state investigate, chi è il responsabile della soluzione, quando è stata implementata e dove è stata svolta. Chiaramente deve essere chiaro il perché è stata scelta questa soluzione, e se la stessa è riuscita a risolvere il problema oppure no. Nelle situazioni reali la scrittura di questa sezione avviene nelle prime fasi, e potrebbe accadere di prendere in considerazione cause meno utili di altre: in questo scenario s'indicheranno delle soluzioni che alla fine del progetto non saranno prese in considerazione (proprio per questo motivo è indispensabile gestire le diverse revisioni del documento e archiviare solamente l'ultima versione).

2.1.5.7 Check of Effects

In conformità a quella che la metodologia PDCA è indispensabile inserire una sezione per la valutazione dei risultati delle soluzioni proposte. È realmente importante costruire queste sezioni: trovare una soluzione e implementarla senza avere fatto dei rigorosi controlli sull'effettiva bontà della soluzione, potrebbe portare a una perdita di tempo.

Grazie a questo tipo di verifica è possibile capire se la soluzione è riuscita a raggiungere gli obiettivi, e permette di stabilire un collegamento tra l'azione implementata e l'effetto prodotto. È quindi importante capire le relazioni che legano le cause con gli effetti e non solo implementare una serie di soluzioni per riuscire a risolvere il problema.

2.1.5.8 Follow-up plan

L'ultima sezione dell'A3 traduce in pratica la fase ACT del PDCA. Attraverso la definizione del piano di azione è possibile identificare e pianificare le azioni che s'intendono eseguire per risolvere il problema.

Capitolo 3

3.1 Trade-Off Curves

Per riuscire a decidere tra una serie di possibili soluzioni alternative tra di loro, Toyota e i suoi ingegneri usano uno strumento relativamente semplice ma con un grande potenziale: le curve di trade-off.

L'utilizzo di questo strumento permette l'analisi simultanea di diverse soluzioni, l'aggregazione di una quantità notevole d'informazioni in un solo grafico, e rende possibile la condivisione e la diffusione della conoscenza all'interno dell'azienda.

Toyota, benché utilizzi costantemente le curve di trade-off per prendere decisioni relative ai propri progetti, non fu la prima a utilizzarle. Il primo utilizzo fu per opera dei fratelli Wright durante i loro studi per la creazione del primo aeroplano al Mondo. Sono tuttora considerati gli inventori dell'ingegneria aeronautica e il loro approccio all'ingegneria differiva di gran lunga rispetto a quella del periodo, che prevedeva, e nella maggior parte prevede ancora, due fasi ben distinte: la progettazione e quindi il test di quello che si era progettato. Con quest'approccio, quindi, inizialmente si fanno degli studi approfonditi sull'argomento, si crea il progetto dopodiché si fanno delle prove per essere sicuri dell'efficacia della soluzione progettata: se i test vanno bene si è sicuri dell'efficacia di quanto progettato (ma non si è sicuri se la soluzione trovata sia la migliore), mentre nel caso in cui i test vadano male, bisognerà tornare indietro con il lavoro e modificare quanto progettato. Chiaramente il processo così descritto potrebbe andare avanti per sempre, o comunque potrebbe non convergere mai verso la soluzione migliore. L'approccio dei fratelli Wright si basava sul testare e poi progettare.

I due fratelli prima di iniziare a progettare eseguirono numerosi test su tutte le parti dell'aeroplano, e furono i primi a utilizzare le gallerie del vento per eseguire test di natura aerodinamica. Oltre ad introdurre molti concetti dell'ingegneria moderna, furono i primi a utilizzare delle semplici tabelle in cui annotare tutti i dati dei loro test. Non solo riuscirono a raccogliere una quantità impressionante di dati, ma riuscirono a capire come i dati cambiavano al variare, e quindi com'erano influenzati, di determinati parametri dei sistemi che testavano. Il passo successivo alla raccolta dei dati era la loro rappresentazione e visualizzazione sotto forma di grafici rendendo così possibile una loro più facile comprensione e analisi.

Il risultato fu l'introduzione delle così dette curve di trade-off.

Il punto più importante negli studi dei fratelli Wright fu quello dell'introduzione di una precisa e organizzata maniera per la generazione e la gestione della conoscenza nell'ambito della progettazione. Durante i loro studi, infatti, non solo crearono un nuovo modo di progettare, ma anche un nuovo e importantissimo modo per la generazione e l'utilizzo delle informazioni e delle conoscenze, che poterono essere utilizzate per progettazioni future.

S'incomincia dunque a parlare di learning cycle in cui si esplorano le diverse possibilità (i set di soluzioni) per la risoluzione di un determinato problema di progettazione, e la riduzione di questo set per riuscire ad arrivare a una soluzione efficace ed efficiente, rendendo possibile scoprire cosa funziona e, cosa ancora più importante, cosa non funziona.

Questo strumento, particolarmente importante per la riuscita degli studi dei fratelli Wright, scomparso dall'uso comune della progettazione ingegneristica dopo la fine della seconda

guerra mondiale. Durante i due conflitti mondiali, le trade-off curves, furono uno strumento molto utilizzato nel campo dell'ingegneria aeronautica.

Uno degli esempi più emblematici della potenza di questo tipo di strumenti fu la progettazione di un aereo da combattimento chiamato "superb fighter". Tale aereo aveva una potenza di 1200 cavalli con una velocità di punta di 630 Km/h. Gli ingegneri riuscirono a progettarlo e a costruirlo (dall'ideazione alla produzione) in soli sei mesi. Quando fu chiesto al capo progetto come avesse fatto in così poco tempo a progettare e produrre un sistema così complesso, egli rispose che uno dei punti di forza della progettazione era la conoscenza dei trade-off per tutti gli aspetti di maggior importanza del sistema. Una volta costruite le curve, bisognava solamente scegliere i valori dei parametri in relazioni ai risultati che si volevano ottenere, e progettare di conseguenza.

Questo strumento scomparso totalmente dalla produzione americana (sia industriale che aeronautica).

Molti studiosi oggi si domandano come sia possibile che questo strumento sia scomparso da un giorno all'altro. Molti rispondono citando l'aumento della complessità dei progetti. Uno dei maggiori esperti di Lean Thinking e di trade-off curves, A.Ward, indica, come possibile causa, il processo. Secondo il suo punto di vista non ci sarebbe un diretto collegamento con l'aumento della complessità del progetto, bensì con il processo nel quale il progetto è inserito.

Oggi i computer e le simulazioni permettono di condurre esperimenti senza correre rischi. Quello che lo studioso sottolineava e la differenza sostanziale tra una simulazione e un test. Uno degli esempi citati è quello relativo alla progettazione e produzione del boeing 777: il boeing fu interamente progettato e costruito, con enorme successo, attraverso simulazioni a scapito, però, di un grosso sforzo in termini di tempo e di costo.

I concetti portanti avanti parecchi anni prima dai fratelli Wright, furono introdotti in Toyota dopo la seconda guerra mondiale, quando numerosi ingegneri dell'aeronautica si trovarono senza occupazione e furono assunti nella casa automobilistica Giapponese. Molti di questi ingegneri avevano appreso le tecniche di progettazione dei fratelli Wright durante il loro lavoro nella progettazione di aeroplani militari usati nella seconda guerra mondiale. Quando iniziarono la progettazione di automobili all'interno di Toyota portarono con loro tutta la conoscenza che avevano accumulato durante la loro precedente occupazione. Durante gli anni successivi Toyota fu in grado di raccogliere un numero impressionante d'informazioni legati ai progetti che nel frattempo venivano portati avanti. Per raccogliere queste informazioni si utilizzavano strumenti diversi: checklist, A3 reports, limit curves, trade-off curves.

Uno dei concetti più importanti da sottolineare è la reale importanza di avere le informazioni raggruppate in forma visuale. Tutti gli strumenti precedentemente citati, infatti, sono strumenti visuali. Mostrare una tabella con una serie di numeri per una serie di variabili, non ha la stessa potenza e semplicità di comprensione che vedere gli stessi dati per le stesse variabili organizzate per esempio in un grafico.

Esiste un detto che dice: "Le descrizioni scritte sono migliori delle descrizioni orali, i numeri sono meglio delle parole, e le immagini sono meglio di tutti".

Attraverso questo strumento Toyota è in grado di raccogliere e analizzare un numero molto grande d'informazioni e di conoscenza e di utilizzarle per progettare nuovi sistemi nella maniera più precisa possibile, senza dover ricorrere a successive modifiche o riprogettazioni. Questo strumento è particolarmente utile quando si vogliono istruire nuovi ingegneri (i quali per esempio possono essere stati appena inseriti nel team che si occupa di

un determinato progetto) o quando si vogliono spiegare determinate scelte a clienti o fornitori.

Le trade-off curves non sono altro che una serie di curve costruite in maniera tale da legare due o più variabili (parametri del sistema). Così facendo si possono osservare in maniera molto semplice quali sono i trade-off nella scelta dei valori per le variabili considerate, e come vari il valore di un parametro al variare dell'altro. Il concetto che non deve essere trascurato e fatto passare in secondo piano, è la possibilità offerta dalle curve di trade-off di generare conoscenza. Per questo motivo le curve devono essere costruite in maniera più generale possibile.

Oltre all'opportunità di accrescere e di comunicare informazioni all'interno dell'azienda, attraverso l'utilizzo delle trade-off curves, si può riuscire a capire meglio il problema da affrontare o quali sono le sue implicazioni tecniche. Nella maggior parte dei casi la scelta dei parametri da utilizzare è un aspetto critico: l'analisi sarà effettuata attraverso di essi, e quindi una scelta errata delle variabili porterà sicuramente a un'analisi sbagliata o a una non completa comprensione del fenomeno oggetto di studio (lo scopo finale è quello di capire come le variabili governano il limite delle performance del sistema che si sta progettando). Un possibile approccio alla scelta dei parametri potrebbe essere quella di combinare differenti variabili in un singolo parametro permettendo così di inserire e analizzare un numero d'informazioni molto superiore in un singolo grafico.

La potenza di questo strumento è dovuta al fatto che si combinano assieme due differenti caratteristiche: da una parte la precisione matematica dello strumento, dall'altro la capacità dell'essere umano di apprendere un numero molto maggiore d'informazione attraverso l'uso della vista.

Esistono due principali approcci alla costruzione delle trade-off curves.

Il primo prevede l'identificazione preliminare di una serie di variabili ritenute importanti che potrebbero essere investigate attraverso le curve. Il secondo possibile approccio prevede, invece, la creazione di una lista di possibili utilizzi delle curve.

3.1.1 Come si costruiscono

Facendo riferimento a un tipico caso in cui si voglia risolvere un problema, si può pensare di costruire un A3 per tenere traccia di tutte le informazioni necessarie e importanti per affrontare il problema, e cercare di risolverlo. Il primo punto è trovare una definizione sintetica del problema e riportare questa informazione in cima all'A3. Il passo successivo è quello di investigare sulle cause che portano alla manifestazione del problema: questo può essere riportato sull'A3 mediante un grafo ad albero, unito a un'illustrazione semplice e chiara del problema (se per esempio si vuole risolvere un problema di rottura di un particolare meccanico, si potrebbe inserire una foto del pezzo rotto). È molto importante capire quali sono i motivi responsabili di fare insorgere il problema, altrimenti si corre il rischio di concentrarsi su piccoli particolari che potrebbero avere una qualche correlazione con il problema, perdendosi nella miriade di dettagli senza riuscire ad aumentare la comprensione del fenomeno.

Una volta capiti quali sono i legami e le relazioni tra i diversi parametri che portano alla generazione del problema, è possibile costruire la curva che lega le variabili: questo procedimento non necessariamente deve essere visto come un procedimento sequenziale. Infatti, la scoperta delle cause del problema porta sicuramente alla costruzione della curva. La comprensione e l'analisi della curva a sua volta porta a una maggiore comprensione del problema. Quest'aumento delle informazioni e della comprensione può portare a una

modifica delle cause e delle azioni pensate per risolvere il problema. Questo ci fa capire come non si è in presenza di un'attività sequenziale, ma piuttosto a un'attività ciclica.

Punto fondamentale per riuscire a costruire la curva sono i dati. Anche in questo caso è possibile pensare a un'attività ciclica in cui prima si costruisce una curva su un modello matematico, dopo di che si raccolgono i dati e si riportano sul grafico, in maniera tale da migliorare la precisione della curva e darle maggiore robustezza.

I dati possono essere raccolti in diversi modi: mediante test, mediante simulazioni e da sistemi analitici. Comunque si raccolgano i dati, il concetto da tenere in maggiore considerazione risiede nel fatto che ogni volta che si hanno a disposizione nuovi dati, questi devono essere usati per aggiornare e migliorare le curve di trade-off. Se non si dà la giusta importanza a quest'aspetto si rischia di perdere il vantaggio stesso nell'uso delle curve, con la reale possibilità che il loro utilizzo diventi anti produttore (analizzare un problema, e prendere delle decisioni su com'è possibile risolverlo, facendo affidamento su dei dati obsoleti, non ha molto senso).

L'importanza e l'utilità di creare delle curve di trade-off trova il suo apice quando le curve sono create mediante dati sperimentali, e cioè provenienti da test reali. Esistono dei casi in cui è però utile creare e analizzare le curve di livello, basando la loro costruzione su formule matematiche. Questo tipo di studi, mediante curve costruite mediante equazioni matematiche, sono particolarmente utili nelle fasi iniziali di progettazione. Per esempio si può riuscire a capire come un determinato parametro è influenzato al variare di un altro parametro senza dover costruire nessun modello o utilizzare un ambiente di simulazione. Le informazioni sono facilmente e immediatamente comprensibili e applicabili dal team di sviluppo.

L'uso di tecniche sperimentali per la raccolta dei dati porta con sé diversi vantaggi:

- I dati sono raccolti “sul campo”
- Spesso si riescono a trovare problemi legati all’analisi delle cause
- Economico, soprattutto nel caso si possano utilizzare prodotti in produzione rispetto alla costruzione di un prototipo

Raccogliendo dei dati sul campo, mediante esperimenti, si ha l’indiscutibile vantaggio della robustezza dei dati stessi nonché della possibilità di capire che succede durante il funzionamento della soluzione e capire possibili cause che non si sarebbero potute rintracciare utilizzando dati storici e simulazioni.

In contrapposizione ai vantaggi, esistono alcuni punti a sfavore di questa metodologia:

- Spesso lenta e costosa (a seconda dei casi)
- Spesso o limitato e irrealistico, o troppo generico e difficile da capire

L’uso delle trade-off curves permette di creare dei grafici prendendo in considerazione una variabile e studiando come il suo valore cambia al variare di una seconda variabile considerata. Questo permette di avere le conoscenze necessarie per progettare una soluzione ottimale che si ha la certezza che sia valida. In buona sostanza questo permette di fare una distinzione tra area di fattibilità e area di non fattibilità. Infatti, prendendo in considerazione una qualunque trade-off curve si può evidenziare come esista, per una data soluzione o progetto, un’area di fattibilità e una di non fattibilità. L’entrata o l’uscita da una di queste aree è legata dal valore assunto dalle due variabili prese in considerazione.

Per comprendere meglio l’approccio a questo tipo di strumento si può analizzare un caso.

Si supponga di essere all’interno di un’azienda manifatturiera e che si voglia risolvere un problema. Si può fare riferimento a un generico problema nella progettazione di un

qualsiasi componente meccanico o manifatturiero. Si pensi per esempio a un'azienda meccanica che ha necessità di capire come un certo tipo di carburante impatti sulle prestazioni e sul funzionamento di un motore che produce per una data regione del Mondo.

La prima operazione che un team può e deve fare è quella di analizzare il problema e cercare di individuarne le cause per riuscire a proporre una soluzione. L'output di questo processo nella maggior parte dei casi è un report. Il report può essere pensato e organizzato in due forme principali: il report classico in cui si ha una prima fase descrittiva in cui si spiega e si crea il contesto in cui il problema è inserito e nel quale si manifesta, seguito da una parte in cui si propone e si analizza una data soluzione che punta a risolvere il problema e come ultima fase si esegue il test di quella soluzione su un prototipo vero e proprio. La parte conclusiva di questo tipo di report è solitamente una spiegazione del procedimento seguito e l'indicazione se la soluzione riesce oppure no a risolvere il problema. Il secondo tipo di report che è possibile elaborare è abbastanza differente: la prima parte, solitamente, è lasciata inalterata, le successive invece non contengono lunghi paragrafi di spiegazione ma soltanto dei grafici.

Una delle caratteristiche del secondo tipo di report è che non sono presenti conclusioni scritte ma soltanto dei grafici. Per la costruzione di questi grafici quello che è importante avere a disposizione sono una serie di dati recuperati da test sul prodotto finito, o su una sua parte, per diversi valori dei parametri presi in considerazione. Se il primo report, e di conseguenza i test effettuati per costruirlo, è stato ideato e creato per dare una risposta precisa a un determinato problema, il secondo report prende in considerazione una serie di test, non solo atti a provare l'adeguatezza di una possibile soluzione, ma ideati per riuscire a creare conoscenza su come può variare un parametro (per esempio le performance di un motore) al variare di un altro a esso legato (per esempio l'uso di un particolare tipo di carburante).

Il primo tipo di test risponde a una specifica domanda sotto specifiche condizioni, nella maggior parte dei casi con un sì o con un no. Il secondo tipo di test, invece, risponde anch'esso alla domanda ma crea della conoscenza riusabile per rispondere ad altri tipi di domande (il primo test non riesce a rispondere ad altre domande perché è stato ideato e creato per rispondere solamente a una specifica domanda).

Da quest'analisi su come l'utilizzo delle trade-off curves possano essere utili quando si vuole rispondere un determinato quesito, e su come riescano a generare una quantità notevole d'informazioni e conoscenza riutilizzabile, fa capire l'importanza nell'utilizzo di un nuovo paradigma di progettazione: non più “design then test”, bensì “learn then design”.

Per le aziende che vogliono guadagnare dei vantaggi nell'uso delle tecniche di lean development i punti su cui focalizzarsi sono i dati e la conoscenza che è possibile generare. Come si è detto il vantaggio principale nel passare da un tipo di progettazione definito “point-based” a uno invece “set-based”, è il gran numero d'informazioni che è possibile generare, e quindi la grande opportunità per le aziende che applicano questi concetti di accrescere le loro conoscenze, e riuscire ad avere le giuste informazioni al momento giusto nel punto giusto.

La sfida principale è dunque quella di riuscire a definire una metodologia, o uno strumento, che supporti la trasformazione dei dati in conoscenza riutilizzabile, e permettere di ottenere (quasi) sempre dei buoni progetti e quindi dei buoni prodotti.

Oltre al concetto d'informazione, intesa come capacità dell'azienda di apprendere dal lavoro precedentemente fatto e di applicare tale conoscenza anche a progetti futuri, c'è da tenere in considerazione la necessità di qualsiasi azienda, in qualsiasi settore, di comprendere e riuscire a soddisfare al meglio le esigenze del cliente. Mediante questo

strumento è possibile mettere in risalto i punti chiavi di un progetto e scegliere tra diversi di essi in maniera tale da massimizzare le performance più importanti per un dato cliente, o per un gruppo di clienti di riferimento: quello che gli ingegneri devono fare e prendere i valori dei parametri indicati nel grafo a seconda della performance desiderata.

Tutti questi problemi possono essere risolti utilizzando le trade-off curves.

Qualunque sia lo scopo dell'utilizzo di questo strumento, preliminarmente si devono identificare una serie di possibili cause che portano alla generazione del problema. Così facendo si possono identificare clusters di fattori e di variabili che formano gruppi logici. Il passo conseguente è l'identificazione di un gruppo di fattori e di variabili che saranno usati per la costruzione delle curve di trade-off. Una volta che si sono riusciti a recuperare i dati e disegnare il grafico è possibile visualizzare il punto in cui il progetto, o la soluzione a un particolare problema, non rispetta le specifiche e quindi fallisce. Come sottolineato precedentemente è indispensabile capire con precisione quali variabili utilizzare: se s'identificano, per esempio, quattro variabili, diventa impossibile costruire un grafico che le riporti simultaneamente tutte. Si può dunque pensare di sceglierne due, oppure di tenerne fissa una e combinarla con le altre tre. È dunque importantissimo fare diverse prove con diverse combinazioni, così come diventa importantissimo la scelta della scala di valori da utilizzare: per esempio la scelta di una scala logaritmica può essere maggiormente efficace rispetto a una scala lineare.

Fare la giusta scelta delle variabili porta a diversi vantaggi:

1. Fare la giusta scelta riguardo alla combinazione delle variabili permette di visualizzare in maniera più semplice ed efficace un gran numero di dati.

2. Una buona scelta permette di visualizzare e capire i limiti delle performance della soluzione scelta, permettendo quindi di riuscire a stare all'interno di questi limiti (mediante la scelta opportuna dei valori da assegnare ai parametri del progetto)
3. Un'accurata scelta dei parametri permette di chiarificare cosa bisogna fare per migliorare la situazione.

Non esiste una ricetta che permetta di fare la giusta scelta in tutte le possibili situazioni e in tutti i possibili progetti che si andranno ad affrontare. Le condizioni possono variare anche per progetti molto simili, l'ambiente e le circostanze variano, così come le richieste del cliente e i requisiti. L'unico procedimento veramente valido è il miglioramento costante dei progetti e cercare di semplificarli sin quanto è possibile, in maniera tale da focalizzarsi sugli aspetti realmente importanti.

La bontà delle scelte precedentemente descritte si rifletterà sulle prestazioni del lavoro del team, nel momento in cui sarà necessario lavorare mediante le curve di trade-off. È dunque importantissimo riuscire a studiare le curve e analizzarle correttamente per capire il problema, capire i limiti delle soluzioni proposte e capire qual è la soluzione migliore. Non è chiaramente da trascurare il processo di documentazione dei risultati prodotti, in maniera tale che tutta la conoscenza creata possa essere utilizzata in future analisi o progetti futuri.

Riepilogando si può fare una lista dei motivi che portano all'utilizzo delle trade-off curves:

1. Come guida per una buona progettazione: costruendo le trade-off curves, e analizzandole, si viene a conoscenza di dettagli che possono essere usati in maniera rapida ed efficace nel proseguimento del progetto
2. Per capire, addestrare e aiutare a ricordare il personale coinvolto
3. Per valutare i fornitori (per esempio in base alle soluzioni proposte e alla loro efficacia)

3.1.2 Costruire le trade-off curves

Inizialmente si costruisce un team (un piccolo numero di esperti) disponibile nel provare l'efficacia nell'utilizzo di questo strumento. Il team sarà responsabile da tre a quattro cicli di learning focalizzati su un singolo problema. Alla fine di questi cicli il risultato deve essere la generazione di una serie di curve che permette di stabilire come risolvere il problema e quali sono i suoi limiti, ed evitare che si ripresenti in futuro.

Una volta trovata la soluzione è essenziale l'analisi dei risultati ottenuti e la loro documentazione: la documentazione non deve essere un esercizio fine a se stesso, ma deve essere un'esperienza che permetta di capire maggiormente i problemi incontrati e la soluzione proposta, permettendo di migliorare l'uso della nuova metodologia.

Ogni volta che si deve affrontare un nuovo problema o un nuovo progetto è necessario rivisitare le conoscenze pregresse e le informazioni, sotto forma di trade-off curves, accumulate nel corso del tempo. Non solo. Le curve storiche accumulate nel corso del tempo e utilizzate nei progetti attuali possono, e devono, essere costantemente aggiornate per migliorare la loro capacità di creare e diffondere conoscenza all'interno dell'azienda e dei team di progetto.

3.1.3 Esempi pratici di utilizzo

3.1.3.1 Esempio: exhaust system prototypes

Durante un'intervista con un fornitore di Toyota, responsabile della prototipazione del sistema di scarico delle autovetture, fu chiesto quanto nuovi sistemi fossero forniti a Toyota per il prototipo di un nuovo mezzo. La risposta fu: “dieci, forse venti, delle volte siamo arrivati a fornirne cinquanta per un singolo modello, anche se quest'ultimo è stato un caso estremo”.

La domanda più ovvia a questo punto è quella del perché c'era la necessità di progettare così tanti sistemi per un singolo modello di automobili. La risposta fu: “il responsabile degli ingegneri vuole conoscere i trade-off”.

Il fornitore testava i vari sistemi su un motore in prestito da Toyota, in maniera tale che ogni sistema fosse accompagnato da dei dati ricavati da test reali. Questi dati, provenienti da diversi test su sistemi differenti, erano combinati assieme per riuscire a creare le trade-off curves. Toyota utilizzava le curve per fare la scelta sulla migliore soluzione proposta dal fornitore.

La stessa identica soluzione era proposta svariate parti della macchina: freni, parti estetiche e così via.

Questo esempio merita un'ultima considerazione relativa ai costi: progettare e prototipizzare così tanti sistemi, come nel caso appena citato, ha un costo. Nello specifico è possibile creare un numero così elevato di sistemi di scarico, in quanto è relativamente economico e semplice costruire un prototipo per questo tipo di sistemi. Il costo quindi ha una grossa influenza sulla scelta del numero di prototipi e di test da eseguire, e deve essere attentamente tenuto sotto controllo.

1.3.1.2 Esempio: exterior plastic moldings

Una volta che in Toyota sono state fatte le ovvie considerazioni e sono state prese le prime decisioni sull'estetica di un modello di automobili, vengono spedite ai vari fornitori dei disegni "grezzi" delle parti di plastica che dovrebbero essere realizzate. Tali disegni nella maggior parte dei casi sono dei disegni fatti a mano e includono solamente alcuni dati approssimati su altezza e lunghezza.

I fornitori analizzano e studiano il disegno che gli è stato mandato, quindi costruiscono un grafico che mette insieme ogni idea tenendo conto dei materiali, del costo, delle performance e della qualità. La risposta del fornitore è quindi questo insieme di possibili soluzioni (solitamente da cinque a sei). Toyota può così discutere le diverse soluzioni proposte dai diversi fornitori. Una volta che Toyota decide quale fornito è più adatto per quel particolare problema, discutono sui dettagli del progetto come lo stile del pezzo, dove dovrà essere montato e così via, sino ad arrivare a una soluzione ottimale.

In questo caso Toyota acquisisce informazioni e dati dai vari fornitori mediante diverse curve di trade-off. Nel caso in cui Toyota non avesse possibilità di usare questo strumento, la scelta ricadrebbe sul fornitore che già produce il pezzo, o la parte in questione, rispettando, per esempio, i vincoli di costo. In questa maniera però si perderebbe l'opportunità di affidare la commessa a un altro fornitore che può compensare la mancanza di esperienza utilizzando una nuova tecnologia o dei nuovi materiali che li permetta di fornire il pezzo a un costo e/o a una qualità maggiore o uguale rispetto all'altro fornitore.

1.3.1.3 Esempio: il caso Harley Davidson

L'azienda aveva la necessità di lanciare sul mercato una nuova linea di moto. In questa fase trovare dei casi di “false positive feasibility” rischia di far slittare i tempi di consegna o di entrata sul mercato.

Nel caso specifico l'azienda aveva appena terminato la produzione di un nuovo modello. La prima linea di moto fu spedita a una fiera annuale che si svolge in Florida. L'evento cattura l'attenzione di migliaia di clienti vogliosi di vedere i nuovi modelli e di riuscire a provarli per la prima volta. Chiaramente quest'opportunità è colta dalle più importanti aziende produttrici di moto che orgogliosamente portano i loro prodotti di punta per mostrarli ai possibili futuri clienti.

Anche Harley Davidson non voleva farsi scappare le opportunità di questo evento, e quindi decise di portare e mostrare ai clienti il suo nuovo modello. Durante la progettazione ci furono dei problemi per quanto riguarda la cinghia di trasmissione. Nel caso specifico fu scelto di utilizzare come cinghia la stessa che era stata montata nel modello precedente. Sin dai primi test, infatti, il team di sviluppo e di progettazione si accorse che la cinghia di trasmissione non riusciva a superare i test sotto sforzo. Questo dette luogo a numerosi riprogettazioni per riuscire a risolvere il problema.

Il problema, dopo numerosi cicli di riprogettazione, fu risolto e fu data la via libera alla produzione.

Sfortunatamente durante un test su strada, eseguito come dimostrazione pubblica nei giorni in cui si teneva la fiera, la cinghia non riuscì a sopportare lo sforzo e si ruppe. Chiaramente non era possibile applicare le stesse metodologie di riprogettazione che erano

state impiegate nella fase di progettazione e sviluppo, in quanto il team non avrebbe avuto abbastanza tempo prima del lancio sul mercato.

Durante quegli anni all'interno dell'azienda era in atto una profonda rivisitazione delle tecniche di progettazione e sviluppo, orientandosi verso le tecniche lean introdotte da Toyota. Senza capirne il reale potenziale, la squadra ingegneri era stata introdotta all'utilizzo di alcuni strumenti, quali A3 e trade-off curves, che avevano fatto la fortuna della casa automobilistica Giapponese.

Nella situazione di allarme in cui Harley Davidson si trovava in quel momento, si riuscì, però, a capire l'importanza delle trade-off curves e delle conoscenze generate durante la fase di testing. Come si è detto precedentemente, durante la fase di progettazione, la squadra ingegneri si trovò a risolvere non pochi problemi legati all'utilizzo di un particolare modello di cinghia di trasmissione. Questo creò numerosi cicli di riprogettazione, ognuno dei quali prevedeva numerosi test su diversi prototipi in maniera da ricavare dei dati sulle performance della cinghia. I vari test furono eseguiti andando a modificare, durante i diversi cicli, dei parametri, sino ad arrivare a una soluzione che si pensava fosse adeguata. Così chiaramente non fu.

Si tentò, quindi, una strada alternativa, che sino a quel momento non era né mai stata usata, né compresa sino in fondo: le trade-off curves.

I dati erano presenti. C'era solo da capire come renderli facilmente usabili e comprensibili attraverso uno strumento visuale. La costruzione delle curve iniziò cercando di mettere assieme i dati inerenti il numero di miglia della cinghia nel momento in cui si rompeva. La scelta di questi due parametri non portò a risultati concreti.

Furono quindi studiati i parametri legati al carico della cinghia al momento della rottura: successivamente si arrivò a utilizzare un parametro legato alla forza sprigionata dal motore

che causava la rottura della cinghia. Riuscendo a ricavare il punto di rottura si poteva costruire un grafico in cui mettere in relazione il numero di miglie percorse dalla cinghia e la forza erogata dal motore al momento della rottura.

Il team, in un ciclo di learning successivo, costruì un grafico che mostrava la diminuzione della resistenza della cinghia alla forza via via che quest'ultima accumulava miglia. Si eseguirono nuovi test e si annotarono i dati relativi ai parametri individuati. Ci si accorse che il carico trasferito alla cinghia era maggiore rispetto a quello trasferito dal vecchio motore e che, quindi, la cinghia non riusciva a sopportare quella forza.

Come si evince dal grafico, la forza della cinghia degrada più che la cinghia stessa accumula miglia: la differenza rispetto al modello precedente sta nel fatto che la cinghia non degrada al punto da raggiungere il limite di rottura, in quanto il precedente motore aveva un carico massimo inferiore rispetto a quello attuale.

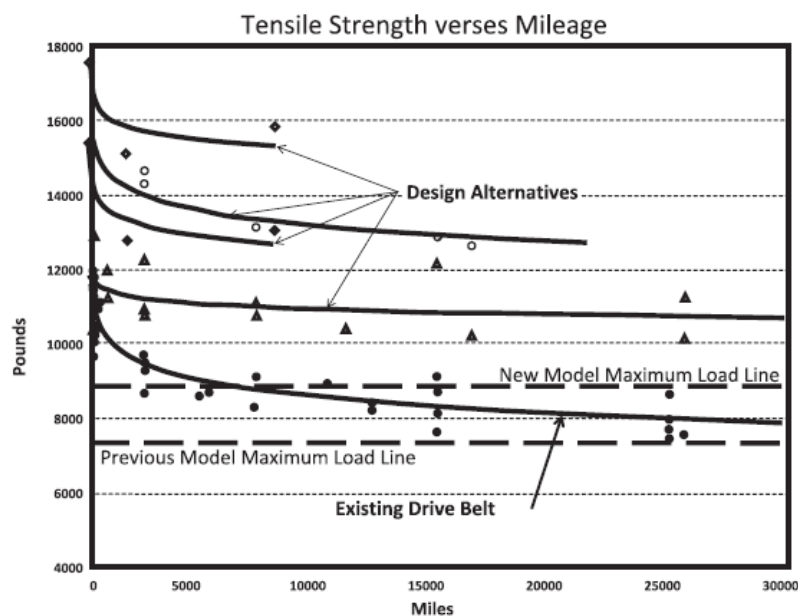


Figura 3.1: Esempio di trade-off curves

Attraverso uno strumento visuale, come sono le trade-off curves, fu abbastanza semplice capire quale fosse il problema. Mediante quest'approccio il team di progettazione non

prese alcun tipo di decisione sul progetto: una volta costruite le trade-off curves per le diverse possibili soluzioni, doveva solamente scegliere i valori ottimali per i diversi parametri chiave in maniera da riuscire a ottimizzare i costi e una serie di altri attributi, con la consapevolezza che il progetto era fattibile.

Ci fu così tanta sicurezza che il progetto non fu testato nuovamente una volta scelti i parametri ottimali, ne furono fatti dei test su strada prima che l'azienda riprese la produzione di quel modello.

Capitolo 4

4.1 Product lifecycle management

4.1.1 Introduzione

Oggi giorno si sta assistendo a un profondo mutamento del mercato in cui le aziende devono competere. Tale cambiamento non riguarda soltanto i prodotti, e la velocità con cui questi devono essere portati sul mercato, ma riguardano anche l'estrema segmentazione del mercato e la velocità di cambiamento della domanda.

Tenendo conto dell'elevata dinamicità del mercato, le aziende non possono che puntare su prodotti sempre nuovi e innovativi, rendendo indispensabile l'uso di un supporto per la gestione, non solo della progettazione e sviluppo e la conseguente produzione, ma anche di dell'intero ciclo di vita del prodotto, che parte dall'idea iniziale sino alla sua dismissione.

L'evoluzione dell'ICT ha permesso la proliferazione di strumenti informatici per il supporto di questo tipo di attività, portando sul mercato software sempre più evoluti e flessibili, che oggi rientrano sotto il nome di PLM (Product Lifecycle Management).

4.1.2 PLM

Il product lifecycle management può essere considerato come il metodo più efficace per la gestione delle informazioni legate ai prodotti all'interno di un'azienda, durante il loro

intero ciclo di vita. Questo tipo di gestione parte dalla prima bozza d'idea sino ad arrivare al momento in cui il prodotto è ritirato dal mercato.

Quando si parla di prodotto per un'azienda, si può far riferimento a un duplice significato: prodotti tangibili, e prodotti intangibili (come un software o un servizio erogato).

Un prodotto, nel significato classico del termine, è un oggetto fisico, tangibile che può essere venduto e distribuito in differenti paesi, nazioni o stati, senza cambiare la propria identità o caratteristiche. Ai giorni nostri il termine prodotto può anche riferirsi a un oggetto immateriale come un modulo di un software, un'informazione, un algoritmo o un servizio.

Molte aziende producono e commercializzano prodotti molto complessi che richiedono l'assemblaggio di una vasta gamma di componenti; ad aggiungere maggiore complessità a questo scenario c'è da considerare i casi in cui i prodotti altamente complessi sono personalizzati su misura per il cliente finale. In questo contesto l'azienda si troverà a gestire una grossa quantità di dati derivanti, per esempio, dalla definizione di ogni singolo componente facente parte del prodotto finale, dalle specifiche dei diversi prodotti, dal progetto del particolare pezzo o prodotto e così via. Questi sono solo alcuni degli esempi che è possibile fare per riuscire a rendersi conto della complessità nella gestione dei dati legati alla progettazione e sviluppo di un prodotto.

Oltre a queste informazioni e dati c'è da tenere in considerazione la sempre maggiore tendenza all'interno delle aziende di sviluppare un ambiente totalmente connesso, in cui le singole funzione aziendali (R&D, marketing, financial...) collaborano e lavorano a stretto contatto senza il bisogno di dover condividere la stessa macchina o anche lo stesso edificio di lavoro. Tutte le parti interessate devono avere accesso all'ultima versione di un

particolare documento e tutti devono essere messi al corrente nel caso in cui sia fatta una modifica a un file condiviso.

Solitamente questa serie d'informazioni sono salvate e utilizzate in formato digitale rendendo possibile l'implementazione e l'utilizzo di un product lifecycle management.

Gli scopi dell'utilizzo di un PLM non risiedono solamente nella possibilità di condividere, salvare o ricercare oggetti e documenti, ma includono l'analisi dei risultati per un particolare prodotto, test, informazioni sull'ambiente, standard di qualità, requisiti ingegneristici, cambi ordini, procedure manifatturiere, informazioni riguardanti le performance dei componenti o dei prodotti e tanti altri.

Uno degli aspetti di maggiore rilievo nei moderni PLM è la possibilità offerta da questi ultimi di gestire il workflow e la messa a disposizione di strumenti che permettono di standardizzare, automatizzare e velocizzare le operazioni di product management.

Facendo riferimento alle parole di Michael Porter sulla value chain ("The idea of the value chain is based on the process view of organizations, the idea of seeing a manufacturing (or service) organisation as a system, made up of subsystems each with inputs, transformation processes and outputs. Inputs, transformation processes, and outputs involve the acquisition and consumption of resources - money, labour, materials, equipment, buildings, land, administration and management. How value chain activities are carried out determines costs and affects profits.") è possibile collegare tutte le aziende che fanno parte della value chain in maniera più rapida, mettendo a disposizione strumenti di ricerca d'informazioni avanzate, strumenti per la condivisione delle informazioni, riuso dei dati e tante altre possibilità, integrando la tracciabilità dei dati e strumenti per garantire la sicurezza dei dati.

Oltre ai vantaggi su citati c'è da considerare i vantaggi in termini di costo: molte aziende, infatti, riportano di un abbattimento dei tempi di pay-off e una sostanziale riduzione dei costi di sviluppo del prodotto.

Alla fine degli anni '80 emersero, nelle aziende manifatturiere, i sistemi EDM (Engineering Data Management) e PDM (Product Data Management). Questi sistemi emersero per dare la possibilità di gestire la grossa mole di dati generata dai sistemi CAD (Computer Aided Design). Attraverso i sistemi PDM si dava dunque l'opportunità di standardizzare gli oggetti, di salvare e controllare i files generati, mantenere e controllare le revisioni dei BOM (Bill of Materials) e non ultimo come importanza identificare con immediatezza e facilità relazioni tra parti di uno stesso prodotto.

Come si è detto in precedenza questo tipo di sistemi nacquero in un contesto in cui il numero di files e di documenti cresceva a dismisura, rendendo necessario uno strumento per una gestione centrale di questa mole di dati: i sistemi PDM permettevano in maniera abbastanza semplice di fare ricerca sui documenti, sulle strutture dei BOM riducendo, al tempo stesso, il rischio di utilizzare versioni incorrette di progetti e aumentando il riuso delle preesistenti informazioni sui prodotti.

Per una definizione precisa di cosa s'intenda per PDM si può fare riferimento alle parole utilizzate da CimData (una delle più importanti aziende di consulenza):

“Product data management is a tool that helps people manage both product data and the product development process. PDM systems keep track of the masses of data and information required to design, manufacture or build, and the support and maintain products – whether your product is an aeroplane, petrochemical plant, highway, railway system, pharmaceutical, automobile, consumer product or service. PDM is used effectively in a multitude of industries.”

L'attività che tipicamente è svolta grazie all'utilizzo di un sistema PDM non è particolarmente nuova. Tradizionalmente i documenti testuali e i disegni sono stati archiviati in fogli di carta: con la crescita esponenziale dei progetti e dei prodotti sviluppati dalle aziende la gestione di questo tipo d'informazioni, con il metodo tradizionale è divenuta impraticabile. Con lo sviluppo informatico sempre più informazioni, dati e disegni sono stati creati e organizzati in maniera digitale, rendendo al contempo molto più semplice l'archiviazione e il recupero delle informazioni necessarie.

Inizialmente i software PDM furono introdotti per dare modo alle aziende di gestire le grosse quantità di disegni CAD che erano generati in fase di progettazione: i PDM venivano sviluppati o come software a se stanti o come moduli separati degli stessi software CAD. In molte aziende, soprattutto nei primi anni in cui fu introdotto, i PDM venivano utilizzati come file system avanzati, senza sfruttare a pieno le grandi potenzialità di questi tipi di sistemi.

I PDM permettono di supportare tutte le fasi durante la progettazione di un nuovo prodotto sino a considerare le fasi in cui sono svolte le attività di supporto.

Come detto in precedenza i PDM furono introdotti nelle aziende manifatturiere, ma via via trovarono largo uso anche in altri tipi di aziende.

Uno degli esempi più famosi d'introduzione e utilizzi di un sistema PDM è quello legato alla FORD. Il progetto d'introduzione di un sistema PDM in FORD fu chiamato C3P. FORD comprende diverse brand al proprio interno, tra cui FORD, Mazda, Lincoln, Mercury, Jaguar, Aston Martin. A oggi possono essere portati avanti in parallelo sino a 60 diversi progetti. Storicamente l'azienda aveva sviluppato una propria soluzione interna per la gestione dei dati dei prodotti, ma con l'aumento delle informazioni, dei dati e dei

progetti, FORD, ha deciso di sviluppare un nuovo ambiente per la progettazione dei prodotti.

Oggigiorno si è passati dal concetto di PDM al concetto di PLM, che permette di includere nella gestione del prodotto, tutte le sue fasi, dall'ideazione alla vendita al cliente, sino ad arrivare all'assistenza post vendita.

I sistemi EDM (Enterprise data management) sono anch'essi nati per permettere alle aziende di avere uno strumento per la gestione di un numero sempre crescente di dati. La gestione attraverso questo strumento permette di avere un controllo dei dati allineandola alla visione, ai principi e ai valori presenti all'interno dell'organizzazione, non che di allinearla agli obiettivi strategici dell'organizzazione. Oltre a questi requisiti un sistema EDM permette di gestire i dati in maniera integrata all'interno dell'organizzazione e di identificare in maniera immediata i rischi e problemi legati alla loro gestione.

Tornando all'analisi delle diverse fasi in cui un prodotto può trovarsi durante il suo ciclo di vita, si possono considerare le seguenti fasi: sviluppo dell'idea, progettazione del nuovo prodotto, sviluppo e commercializzazione.

A completamento di queste fasi principali si può aggiungere la fase di assistenza post vendita. Molte aziende, infatti, cercano di trovare nuove opportunità di business nei servizi specialmente per quello che riguarda l'assistenza post vendita.

Tradizionalmente le aziende manifatturiere sono interessate nell'offrire ai propri clienti una vasta scelta di prodotti. L'obiettivo odierno è quello di offrire anche un servizio che permetta di coprire l'intera durata del ciclo di vita dei prodotti commercializzati, che può durare anche oltre i 30 anni.

La gestione del ciclo di vita del prodotto permette di identificare cinque fasi. In ognuna di esse il prodotto si troverà in un differente stato. Durante la fase denominata “Imagination”, il prodotto è soltanto un’idea nella testa delle persone; la fase successiva, “definition”, l’idea è convertita in una descrizione dettagliata; nella fase “realisation” il prodotto esisterà nella sua forma finale e potrà essere utilizzato dall’utente finale. Le ultime due fasi sono supporto e dismissione.

Ognuna di queste fasi deve essere gestite in maniera opportuna dall’azienda, e ognuna di esse dovrà avrà una gestione separata, con tecniche e strumenti differenti. Nelle prime due fasi, quando il prodotto non esiste fisicamente, è molto complicato applicare le giuste tecniche di gestione, così come sarà ancora complicato gestire la fase di supporto, in quanto il prodotto sarà in mano al cliente.

Potrebbe capitare che l’azienda perda il controllo di una o più fasi del ciclo di vita del prodotto: questo potrebbe portare conseguenze gravi, come l’allungamento dei tempi di consegna al mercato, o al cliente, del prodotto finale, o l’aumento dei costi di produzione e progettazione. Il primo di questi inconvenienti comporterebbe una perdita d’immagine, la seconda una perdita di denaro.

Quello che sta al centro di qualsiasi software PLM è il prodotto. I prodotti sono quello che i clienti comprano, e quindi sono la fonte dei guadagni per l’azienda. Senza i prodotti non ci potrebbero essere i clienti, e senza i clienti non ci potrebbe essere un profitto per l’azienda.

Il PLM permette di gestire i prodotti e di tenerli sotto controllo. Oltre a questo permette di gestire in maniera efficiente ed efficace lo sviluppo di nuovi prodotti, diminuendo i tempi del *time to market* e, al tempo stesso, permettendo il successo competitivo dell’azienda (se

un prodotto è portato sul mercato in un tempo eccessivo, il cliente potrebbe decidere di scegliere un'altra azienda).

L'impatto nell'utilizzo di un PLM coinvolge chiunque lavori allo sviluppo del prodotto migliorandone le performance, aumentando il controllo, riducendo il rischio, e permettendo di avere una vista a 360 gradi di tutto ciò che accade al prodotto.

IL CEO (Chief Executive Officer) utilizza il PLM aspettandosi di aumentare il profitto portando il prodotto sul mercato in tempo limitato (e più velocemente rispetto alla concorrenza): il suo scopo è di avere un completo controllo del prodotto, in maniera tale da evitare sorprese future. Oltre a queste funzioni, il CEO può avere un completo controllo del portfolio prodotti dell'azienda, permettendoli di prendere decisioni su dati reali.

Anche il CFO (Chief Financial Officer) può trarre grossi vantaggi nell'utilizzo di questo tipo di software: può tenere sotto controllo tutti gli aspetti finanziari per un dato prodotto, permettendoli di ridurre il costo per quelle fasi che non aggiungono valore, il valore del portfolio prodotti può essere costantemente monitorato, è possibile fare stime finanziarie per gli shareholders basate su dati reali, creare scenari reali in cui analizzare gli effetti finanziari di un abbassamento dei tempi di produzione di un prodotto, aumentare la propria competitività ecc.

Il PLM dall'opportunità al CPO (Chief Product Officer), la persona che è responsabile per l'intero portfolio di prodotti, di conoscere esattamente lo stato di ogni parte del prodotto, e l'esatto stato delle strutture di ogni prodotto. Il CPO può, quindi, tenere sotto controllo entrambe le fasi di progettazione e di produzione.

Attraverso l'analisi di queste caratteristiche si capisce come il PLM metta nuovamente il prodotto al centro della strategia aziendale, dando la possibilità ai managers aziendali di avere una visione integrata dei loro prodotti, e dei progetti associati al prodotto.

Il valore aggiunto nell'uso di questo tipo di software non è confinato solamente per particolari tipi d'azienda. Al contrario può essere utilizzato da chiunque: aziende manifatturiere, farmaceutiche, food and beverage, software house, aziende di moda ecc. In pratica tutte le aziende che gestiscono un prodotto, sia esso materiale che immateriale, potrebbero trarre grossi vantaggi nell'utilizzo di questo strumento.

Utilizzando un PLM si ha l'opportunità di implementare un sistema per il controllo degli accessi a certe informazioni. Può essere, infatti, necessario limitare l'accesso in maniera tale che gli utenti possano soltanto leggere i dati senza modificarli, oppure impedire l'accesso in base alla tipologia d'utente.

Altre caratteristiche dei PLM sono: l'archiviazione, la fruibilità (rendono disponibili i dati in qualsiasi momento e in qualsiasi luogo), la modifica (permettono la gestione delle modifiche dei dati durante l'intero ciclo di vita del prodotto, permettendo di gestire le possibili diverse fasi delle modifiche dalla richiesta formale all'approvazione) e la gestione in maniera sicura delle diverse copie di uno stesso documento (se esiste più di una copia, la modifica di una delle copie non si propaga sulle altre).

Un aspetto da non sottovalutare è la definizione dei dati. Quando non esiste una definizione standard per un dato, sia esso un prodotto, un componente o una soluzione tecnica, ogni utente potrebbe avere una propria definizione per l'oggetto in questione. Se non è eseguita una gestione di questo tipo di problemi, gli utenti potrebbero trovarsi confusi nel capire di cosa si parla, creando perdite di tempo quando queste informazioni sono trasferite tra funzioni diverse della stessa azienda, o quando persone di funzioni diverse sono messe a lavorare assieme.

Un'altra caratteristica dei PLM è la tracciabilità del prodotto. Come abbiamo detto le funzioni base dei PLM sono la gestione degli oggetti, dei documenti, delle strutture e delle

modifiche al prodotto. Tutte le modifiche che sono state eseguite al prodotto, al piano o al progetto, le ragioni che hanno portato alle modifiche e chi le ha proposte, sono tutte registrate in un unico ambiente e in un unico database, quello del PLM.

Normalmente un numero considerevole d'informazioni relative a un prodotto e alla sua produzione, al testing sono raccolte durante l'industrializzazione del bene. Tuttavia molte informazioni sono racchiuse all'interno delle aziende o anche all'interno delle funzioni aziendali, o peggio ancora all'interno di diversi sistemi informativi. È quindi necessario collegare ogni informazione necessaria per realizzare la tracciabilità del prodotto.

Quello che è ancora abbastanza difficile da gestire è la tracciabilità del prodotto all'interno della supply chain, a causa del numero dei diversi sistemi informativi che deve essere integrato. Le informazioni necessarie a questo tipo di tracciabilità sono create durante il trasporto, la distribuzione, la ricerca dei componenti e durante la manifattura e la manutenzione del prodotto.

Chiaramente all'interno di qualsiasi supply chain saranno presenti diverse aziende operanti su diversi aspetti della fabbricazione del prodotto: queste aziende devono essere in grado di scambiarsi informazioni per riuscire a tracciare i prodotti.

4.1.3 Lifecycle

L'idea di associare un lifecycle ai prodotti esiste da moltissimo tempo all'interno delle industrie e delle aziende. Questo concetto è stato usato prevalentemente nelle industrie che producono prodotti tangibili, come aerei, che hanno un ciclo di vita molto lungo; quello

che sfugge ancora oggi a molte aziende è l'importanza della gestione del ciclo di vita una volta che il prodotto è uscito dalla fabbrica.

Il ciclo di vita di qualsiasi prodotto (sia tangibile sia intangibile) può essere visto secondo diversi punti di vista, a seconda della funzione aziendale: per esempio secondo il punto di vista della funzione marketing il ciclo di vita è orientato al marketing (si potranno avere quattro fasi: introduzione, crescita, maturità e declino). Secondo il punto di vista dell'utente il ciclo di vita del prodotto inizia quando lo compra e finisce quando smette di usarlo.

Guardando il ciclo di vita secondo il punto di vista dell'utente o secondo il punto di vista dell'azienda può portare a considerare differenti fasi: le prime tre fasi sono identiche per entrambi gli attori (immaginazione, definizione e realizzazione), mentre le ultime due sono differenti. Per l'utente, infatti, avremmo uso e dismissione, mentre dal punto di vista aziendale avremmo supporto e ritiro. Si va, quindi, incontro a una differente gestione nel tempo di queste fasi: il cliente può smettere di utilizzare il prodotto, ma l'azienda potrebbe continuare a supportare il prodotto per altri utenti; un'azienda potrebbe ritirare un prodotto prima che un utente dismetta lo stesso. Le fasi quindi non avvengono una dopo l'altra, ma si sovrappongono.

Uno dei problemi più importanti per molte aziende è che non si ha una chiara conoscenza di quello che succede alla fine del ciclo di vita del prodotto. Secondo una ricerca, condotta da Goldense Group Inc., solo il 19% delle aziende hanno un programma di ritiro di prodotti obsoleti. Nel restante 81% delle aziende i vecchi prodotti non sono gestiti.

Le attività svolte all'interno del ciclo di vita del prodotto, rendono possibile produrre e supportare il prodotto o il servizio erogato.

Molti anni fa nessuno prendeva in considerazione la gestione del ciclo di vita del prodotto: negli anni '50 e '60 la preoccupazione per le aziende era quella di portare quanti più prodotti possibili sul mercato. Quello che succedeva dopo non era fondamentale, anzi era secondario.

Durante gli anni '70 furono introdotti i primi sistemi per la gestione dei progetti software. Queste nuove metodologie permettevano di suddividere la progettazione in fasi (Waterfall models). Anche l'industria manifatturiera incominciò a utilizzare questi approcci, in cui le diverse funzioni svolgevano il proprio lavoro in maniera isolata, passandosi il lavoro in sequenza. Negli anni '80 s'introdussero le metodologie di concurrent engineering. Negli stessi anni fecero la loro comparsa i primi EDM (Engineering data management).

Negli anni si diffuse internet, e-commerce, il B2B, si introdussero gli ERP e gli EDM vennero evoluti in PDM.

In questa evoluzione dei sistemi utilizzati all'interno delle aziende, si deve aggiungere l'introduzione dei PLM (2001). Prima dell'utilizzo di questo strumento, le aziende, gestivano il ciclo di vita del prodotto in maniera implicita: solo dopo la diffusione dei PLM si capì il reale valore nella gestione del ciclo di vita dei prodotti.

Durante il ciclo di vita del prodotto, diverse persone, come progettisti, venditori o utenti finali, possono lavorare con il prodotto coinvolgendo diverse attività. Quello che ognuno di essi vuole è svolgere queste attività in maniera efficiente ed efficace, e utilizzare la giusta struttura dati. Gli utenti potrebbero voler lavorare avendo a disposizione la lista dei componenti che costituiscono il prodotto, altri potrebbero voler lavorare sulla lista dei componenti necessari per la sua produzione, e così via. Le strutture possono essere realizzate in diverse maniere: a una dimensione o più dimensioni (in questo modo si riesce a gestire un numero d'informazioni ben superiore).

Un esempio potrebbe essere quello di dover lavorare con il BOM (Bill of Materials): questa è la struttura dei componenti che costituiscono il prodotto finale. Il BOM è quindi una struttura gerarchica che mostra i diversi componenti per uno specifico prodotto finito. Chiaramente potrebbero esserci diversi BOM a seconda dell'azienda.

Qualunque sia il tipo di azienda, la progettazione e la produzione di uno specifico prodotto, la gestione di questo tipo di strutture rende possibile acquisire un numero impressionante di dati. Questi dati saranno necessari per progettare, produrre e supportare il prodotto durante il suo ciclo di vita. Oggigiorno la gestione di questi dati può essere fatta utilizzando la tecnologia informatica e i software PLM.

La gestione del ciclo di vita del prodotto non si esaurisce nel momento in cui il prodotto esce dalla fabbrica o dal magazzino dell'azienda: la sua gestione deve continuare sin tanto che questo non viene dismesso o viene ritirato dal mercato. Questo nuovo tipo di gestione rende possibile allungare le fasi della gestione del ciclo di vita. Lo scopo principale delle aziende che fanno questo tipo di gestione, è quello di offrire ai propri clienti un prodotto di qualità con una serie di servizi aggiuntivi costruiti attorno al prodotto.

Una delle possibilità offerte dalle nuove tecnologie è quella anticipare la manutenzione e permettere la diagnostica a distanza. È da considerare che oggi si ha la tendenza a spostare la responsabilità della manutenzione sui fornitori dei componenti o a dare in outsourcing queste attività. Questa tendenza è rafforzata dalla migliore esperienza e conoscenza dei fornitori sui prodotti che essi commercializzano.

L'obiettivo è quindi spostare l'attivazione della manutenzione nel momento in cui si genera il guasto, o il malfunzionamento, a operazioni tese ad anticipare il malfunzionamento, attraverso la diagnosi a distanza. Per permetterlo è necessario raccogliere i dati in tempo reale, analizzando sia il sistema del cliente sia quello del

fornitore. In questo contesto i fornitori dovranno continuamente monitorare le condizioni dei propri dispositivi e reagire tempestivamente.

La preconditione a tutti questi nuovi servizi è l'utilizzazione, da parte del fornitore, delle giuste attrezzature e dei giusti modelli di gestione delle informazioni e dei documenti. Il fornitore dovrà avere la capacità di controllare l'intera documentazione del prodotto che fornisce, la struttura del prodotto (possibilmente corredata della data o del periodo di progettazione e pianificazione). Questo rende possibile conoscere l'hardware e il software di ogni cliente per riuscire a offrire nuovi servizi.

Nel mercato odierno il confine tra la produzione del bene e l'erogazione del servizio sta lentamente scomparendo. Si assiste a un aumento dei volumi di vendita del servizio rispetto alla produzione del bene vero e proprio. Le aziende cercano di aumentare la vendita del servizio allo scopo di aumentare il valore ed essere meno sensibile alle fluttuazioni del mercato. Dall'altra parte, anche il cliente, vorrà sempre più servizi che agevolino l'uso, l'installazione e la manutenzione del prodotto che comprano.

4.1.4 I benefici nel business

La competizione sempre più accesa all'interno delle quali devono operare le aziende nel mercato odierno, spinge le stesse aziende a cambiamenti continui, sia nella strategia, sia nella scelta dei processi. La velocità di cambiamento è quindi molto elevata, mentre si accorcia drasticamente la durata del ciclo di vita dei prodotti che sono portati sul mercato. La quantità di variazione nella struttura dei prodotti è destinata a salire, in quanto sempre più spesso si assiste alla modifica del prodotto per riuscire a rispettare i desideri dei clienti.

I cambiamenti dei processi sono dovuti a diverse cause, tra cui: crescita della competizione, internazionalizzazione del business, nascita di nuove aziende, riduzione del tempo di consegna, meno tempo disponibile per lanciare nuovi prodotti ecc. Tutti questi fattori portano all'aumentare della velocità con il quale le aziende devono cambiare e innovare i prodotti e i processi.

Oltre a questi fattori si aggiungono i cambiamenti dovuti all'ambiente entro il quale l'azienda opera: questo rende difficile reperire le giuste informazioni sul prodotto e la loro gestione. Il grosso problema in questo caso sarà la scarsa disponibilità di sorgenti d'informazioni: molto spesso le informazioni devono essere recuperate da diversi sistemi informativi, o peggio ancora da diversi PC di diversi dipendenti.

Nello scenario attuale molte aziende possono utilizzare diversi sistemi di gestione delle informazioni, come sistemi CAD, ERP, sistemi di gestione delle vendite e così via. Questo non fa altro che rendere lo scenario ancora più intricato, rendendo necessario un grosso sforzo per integrare tutte queste informazioni.

I PLM rendono possibile questa integrazione, e nello specifico rendono semplice ed efficace la comunicazione tra aziende diverse, ma anche la comunicazione interna a una stessa azienda. Il vantaggio competitivo si ha quando il PLM viene usato per lo scambio di file, per la conversione tra differenti formati o per la gestione di file CAD provenienti da diversi sistemi. Questo rende possibile migliorare la qualità delle informazioni e al tempo stesso ridurre gli errori causati da cattive comunicazioni, evitando la generazione d'informazioni incomplete. Investire in questo tipo di software rende possibile ridurre il tipo e il numero d'informazioni non necessarie.

Uno studio condotto da Coopers & Lybrand ha dimostrato che solo una piccola parte del tempo degli ingegneri, all'interno delle aziende, è speso nella progettazione e sviluppo; la

maggior parte del tempo invece è speso nella ricerca delle giuste informazioni, nella distribuzione e, più in generale, nella gestione delle informazioni. Circa il 20% del tempo è speso nel rifare cose che erano già state fatte in precedenza.

È da considerare che i PLM non migliora automaticamente l'efficacia delle operazioni all'interno delle aziende. Questo sistema è solo uno strumento, usato dalle aziende per diminuire la distanza e per fronteggiare le difficoltà che sorgono nel lavoro quotidiano dei suoi dipendenti. Gli impiegati usano questo strumento per intensificare e migliorare il loro lavoro.

Un grosso problema è che diventa molto difficile misurare i risultati ottenuti con l'introduzione del PLM in termini monetari. Si può, però, considerare che il vantaggio immediato nell'utilizzo del PLM sia il risparmio di tempo e il miglioramento, in termini di performance, del lavoro quotidiano. Oltre a questo vantaggio si deve considerare la riduzione del time-to-market e la riduzione del time-to-react (tempo necessario per reagire ai cambiamenti imposti dal mercato). Tutti questi vantaggi derivano da una maggiore qualità e da una migliore efficienza e velocità delle operazioni di ordine e spedizione del prodotto finito.

In precedenza si è fatto riferimento alla possibilità offerta dai PLM, di migliorare la qualità e ridurre il tempo per la condivisione d'informazioni tra aziende diverse. Considerando un tipico scenario aziendale in cui un'azienda deve gestire i rapporti con tutte le aziende della supply chain, la maggior parte del tempo è speso nella gestione e nello scambio d'informazioni tra le aziende. La condizione ideale si raggiunge quando il fornitore e il cliente sono in grado di utilizzare il medesimo sistema informativo. Per raggiungere quest'obiettivo l'azienda deve essere in grado di distribuire le informazioni in maniera sicura ed efficace ai loro partner.

Anche in questo caso però, il solo utilizzo del PLM non rende possibile raggiungere questo scopo: come detto in precedenza il software è solo uno strumento che rende possibile la nascita di nuove opportunità sia a livello operativo sia a livello strategico. È possibile utilizzarlo per migliorare il lavoro quotidiano dei dipendenti, per costruire diverse interfacce per diverse aziende partner e così via.

Le registrazioni delle informazioni provenienti dal lavoro quotidiano rendono possibile fare diverse analisi statistiche e costruire un robusto sistema, basato su dati reali, per prendere decisioni. Questo rende possibile l'identificazione di nuove metriche, o indici, per compiere analisi sui processi interni all'organizzazione o per misurare le performance delle operazioni svolte quotidianamente all'interno dell'azienda.

L'analisi di queste informazioni, e il loro uso, sono fortemente legati ad altre tecniche di raccolta ed analisi dei dati, come Data Mining, Data Warehousing e Data Mart. Quindi un PLM può essere utilizzato come strumento di supporto al processo decisionale, o come entità a se stante, o affiancandolo ad altri sistemi, come quelli citati precedentemente.

4.2 Aras

Aras Corp fu fondata nel 2000, e inizialmente vendeva il proprio software come proprietario. Nel 2007 l'azienda cambiò strategia e decise di rilasciare il proprio software sotto licenza open source: il business dell'azienda si è spostato dalla vendita alla consulenza software.

Aras Innovator è una soluzione software per la gestione del ciclo di vita del prodotto. Il software si basa sul framework .NET e su SQL Server.

Attraverso questa soluzione è possibile gestire, in un unico ambiente, diverse attività che vanno dal program management, passando al product engineering e arrivando al quality planning. Il PLM si basa sul così detto *Aras enterprise application frame work*. Questo tipo di frame work si basa su SOA, e permette la costruzione di soluzioni scalabili e flessibili permettendo di adattare alle diverse realtà aziendali.

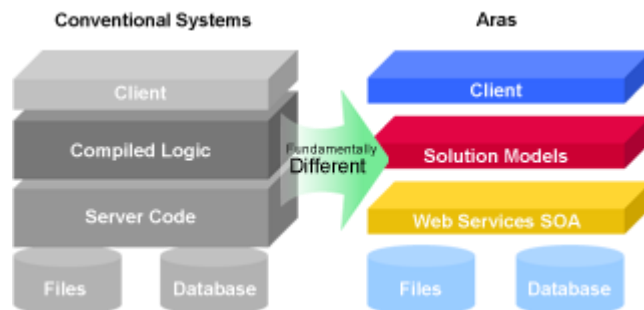


Figura 4.13: Stack Aras Innovator

Una delle maggiori caratteristiche del frame work utilizzato in ARAS è che segue un approccio “model based”, e non il tradizionale “model driven”. Nell’approccio model driven inizialmente si disegna (tipicamente attraverso UML) il modello del sistema e successivamente si ricava il codice sorgente che sarà compilato, linkato, testato, debugato e in fine ne sarà fatto il deploying. Attraverso quest’approccio si corre il rischio che modello e sistema diventino due entità completamente slegate l’uno dall’altro. Con l’approccio model base, al contrario, tutte le modifiche sono fatte sul modello, evitando le possibili differenze tra modello e sistema che quest’ultimo deve rappresentare.

Un altro punto a favore di questo software è l’implementazione delle tecniche SOA. Questo permette di implementare in maniera rapida soluzioni altamente personalizzate, dando la possibilità di modificare soluzioni già esistenti o di crearne di nuove.

Per riuscire a implementare questo tipo di approccio all’interno di ARAS tutto è descritto attraverso l’uso dei meta-data: funzionalità dell’applicazione, business logic, schema dei

dati, schermate, workflows ecc. Ogni item è quindi descritto da un proprio meta-data, e ogni item si registra sul servizio di cui necessita.

A livello generale, quindi, ARAS Innovator può essere descritta come un'applicazione web multi livello con un'architettura service-oriented composta di web clients, application server, database, file server e si basa totalmente su più comuni protocolli di comunicazione, come http, https, xml e soap.

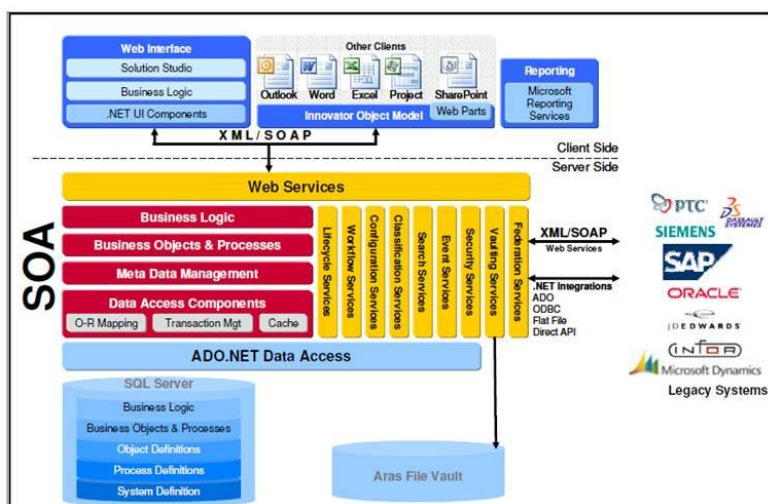


Figura 4.14: Architettura Aras Innovator

Un importante fattore da mettere in evidenza, è che all'interno di ARAS Innovator non è eseguito codice interpretato. L'interpretazione non avviene mai in quanto i modelli sono salvati in schemi predefiniti utilizzati in una serie di Web service. Questi Web service eseguono dinamicamente delle valutazioni dei modelli degli oggetti business.

Attraverso ARAS Innovator è possibile creare in maniera rapida e flessibile delle soluzioni utilizzando una serie di strumenti, quali forms, workflows, file vaulting functions, relationships structures, security access, e a tante altre.

Un altro importante aspetto del software è che i modelli sono definiti completamente in XML e salvati come XML templates. Lavorare mediante XML permette di avere il

supporto di un linguaggio flessibile e facilmente analizzabile, comprensibile e modificabile da chiunque. Questo tipo di modellazione permette una facile condivisione e spostamenti di soluzioni tra un'istanza di ARAS Innovator a un'altra. Tutto quello che c'è da fare è spostare il file XML che descrive la soluzione.

L'approccio flessibile di ARAS Innovator permette una facile integrazione con altri sistemi all'interno dell'azienda. Tale integrazione è resa possibile utilizzando le API messe a disposizione dal programma. Le API mandano e ricevono messaggi di tipo XML/SOAP permettendo di adoperare un'interfaccia standard per lo scambio di dati tra diversi prodotti. Questo permette d'interoperare con altri sistemi IT all'interno dell'azienda, come ERP, sistemi SAP o ORACLE.

Come nei principali PLM, anche ARAS dà grande valore ai documenti. Viene data la possibilità di salvare e ricercare in maniera sicura e semplice tutti i documenti necessari alla corretta gestione del ciclo di vita del prodotto. Oltre alle funzionalità di memorizzazione e ricerca, sono state implementate dei robusti meccanismi per la gestione delle autorizzazioni di accesso ai vari documenti. La semplicità di utilizzo di tali meccanismi per la gestione dei documenti è supportata dalla possibilità di associare documenti, e file in genere, a prodotti o progetti senza dover scrivere neanche una linea di codice, permettendo una rapida e semplice utilizzazione anche agli utenti meno esperti. Altra importante possibilità è la gestione della revisione per ogni singolo documento: ARAS mette a disposizione una serie di funzioni per la gestione delle revisioni, permettendo di tenere traccia in qualsiasi momento delle varie versioni del documento, implementando un meccanismo di controllo delle modifiche.

Il product lifecycle management di ARAS permette di gestire l'intero ciclo di vita di un prodotto, dalla creazione del concetto iniziale, sino ad arrivare alla progettazione,

produzione e ritiro dal mercato. Viene, dunque, data a qualsiasi azienda, di qualsiasi dimensione la possibilità di gestire e di sviluppare processi ripetibili che permettono di ottimizzare l'uso delle risorse, massimizzare i profitti, migliorare la qualità e minimizzando i tempi morti.

4.2.1 ARAS Innovator in dettaglio

Una delle caratteristiche principali da tenere a mente quando si utilizza per la prima volta questo software, è che tutto in ARAS è un ITEM. Quindi quando si crea una form per interagire con l'utente si sta creando un item, quando si crea un metodo si crea un item, quando si crea un documento si crea un item, la stessa interfaccia di ARAS è costruita attraverso gli ITEMS.

Ogni item è descritto da un item type che è a sua volta un item: per ogni item type sarà presente una tabella nel database, in cui per ogni proprietà della item type sarà presente una colonna. Facendo un paragone con la programmazione ad oggetti, è possibile vedere gli item type come le classi e gli item come le istanze di quelle classi. Ogni classe avrà le sue proprietà e i suoi metodi, e quindi anche gli item avranno delle proprietà (alcune sono predefinite dal sistema, ma viene lasciata la possibilità di aggiungere quante più proprietà si vuole) e dei metodi.

Per riuscire a descrivere ognuno di questi item è stato introdotto un nuovo linguaggio, chiamato AML, che sostanzialmente è un dialetto di XML.

AML (adaptive markup language) è un linguaggio flessibile usato per la comunicazione tra client e server.

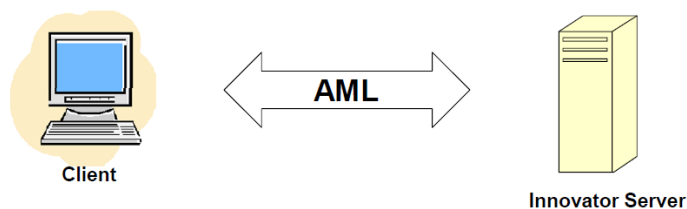


Figura 4.15: Comunicazione tra client e server mediante messaggi AML

Ogni transazione che avviene tra client e server è eseguito mediante uno scambio di messaggio in formato AML; oltre a questo, AML permette di rendere possibile l'integrazione con altri sistemi IT, o di fare interrogazioni sul data base. Sarà possibile ricercare, aggiungere, modificare, cancellare una versione di un documento, fare locking / unlocking di un item e così via.

Un concetto fondamentale quando si analizza il comportamento e le funzionalità offerte da ARAS è la possibilità di creare relazioni tra item. Anche in questo caso la relazione tra due diversi item sarà descritta attraverso un item, che nello specifico prende il nome di relationship item: esisterà un oggetto relationship item per ogni connessione esistente tra due items diversi. Per ognuna di queste relazioni esisteranno due proprietà importantissime: `source_id` e `related_id`, contenente rispettivamente l'identificativo della item sorgente e l'identificativo dell'item riferito.

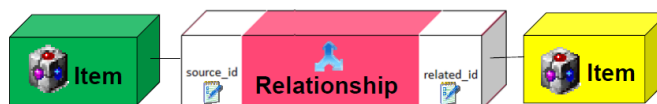


Figura 4.16: Relazioni tra items

Attraverso l'utilizzo del linguaggio AML è possibile costruire complesse query per interagire con il server: sarà possibile ricerca item in base al tipo, o in base alle relazioni cui fanno capo, fare delle ricerche attraverso operatori booleani e così via.

Com'è stato detto in precedenza ARAS si basa sulla metodologia SOA e dunque utilizza SOAP per definire i messaggi tra web service. Anche lo scambio di messaggi AML tra client e server vengono incapsulati, in maniera trasparente all'utente, in un messaggio SOAP.

È possibile fare cambiamenti in tempo reale a qualsiasi soluzione all'interno del software, alle regole di business o alle form che permettono la comunicazione con l'utente finale, senza dover scrivere eccessivo codice. Un'altra caratteristica molto importante è l'estrema facilità di creazione di workflows e cicli di vita.

Tra le soluzioni proposte dal software PLM di Aras si possono citare: BOM management, CAD management, Document management, Engineering change management, Microsoft Office connector, Product analytics, Product costing, Product engineering ecc.

Capitolo 5

5.1 Implementazione in ARAS Innovator

5.1.1 Introduzione

Come si è avuto modo di spiegare nei capitoli precedenti, avere a disposizione certi tipi di metodologie, aiuta la progettazione, rendendola più flessibile, più veloce e più robusta. Negli stessi capitoli si è avuto modo di rilevare la grande importanza che oggi giorno riveste il software PLM all'interno delle aziende, sia come supporto alla progettazione e produzione, sia come strumento per la condivisione d'informazioni.

La fase successiva del presente lavoro di tesi è stata quella di implementare le metodologie precedentemente descritte (gestione rischi, A3, Trade-Off curves) su un software PLM, e la scelta di tale software è ricaduta su ARAS Innovator.

Per raggiungere gli obiettivi della tesi sono state costruite una serie d'interfacce, aggiuntive a quelle già a disposizione del programma, permettendo all'utente di creare nuovi progetti, gestire i rischi associati a esso, generare nuovi learning cycle, per la mitigazione dei rischi, mediante documenti A3 e visualizzare le performance delle soluzioni trovate tramite le trade-off curves.

Per la creazione delle interfacce ci si è serviti degli strumenti messi a disposizione dal programma, e di una serie di file esterni per la generazione delle curve di trade-off.

5.1.2 Modulo PLD (Product Lean Development)

Prima di spiegare in dettaglio com'è stato implementato il modulo PLD, è giusto fare una rapida panoramica sull'interfaccia del programma e come questa si presenta all'utente.

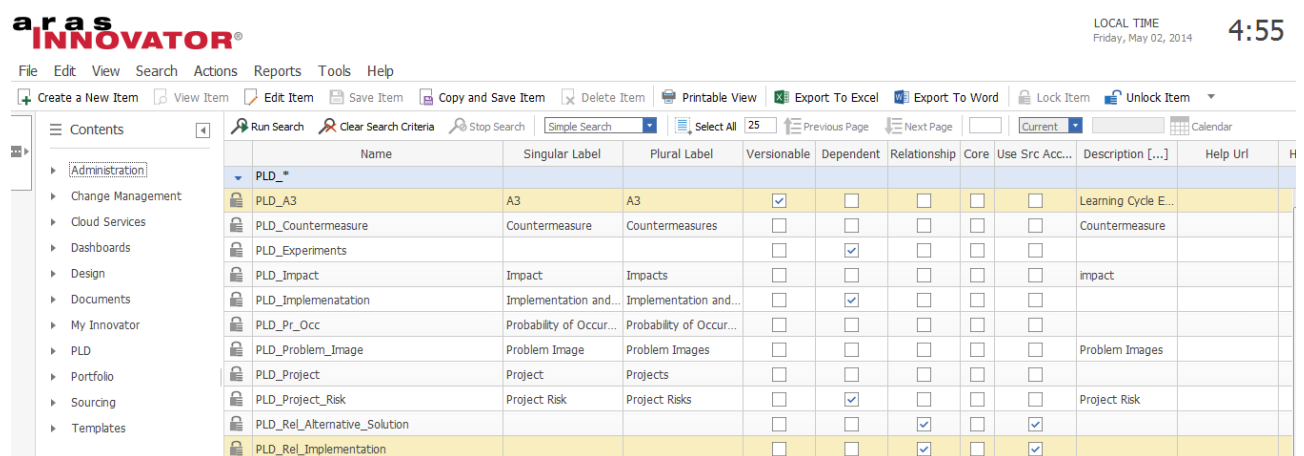


Figura 5.17: Finestra principale di Aras Innovator

L'immagine mostra la tipica interfaccia presentata all'utente, una volta che esso ha immesso le credenziali di accesso. Tale interfaccia può chiaramente variare secondo i privilegi posseduti dall'utente: se per esempio l'utente non è amministratore, non sarà mostrato la voce del menù "Administrator".

Il menù a sinistra della schermata è chiamato TOC e contiene tutti i comandi che è possibile selezionare dall'utente: da qui sarà possibile visualizzare i diversi tipi di items presenti, crearne di nuovi (se si possiedono i privilegi), creare o visualizzare i file CAD o i file BOM e così via. Per ognuno degli item type creati è poi possibile specificare il punto di visualizzazione all'interno del TOC.

Nella figura precedente è possibile visualizzare, all'intero del TOC, la voce PLD: tale voce permette di accedere al modulo implementato.

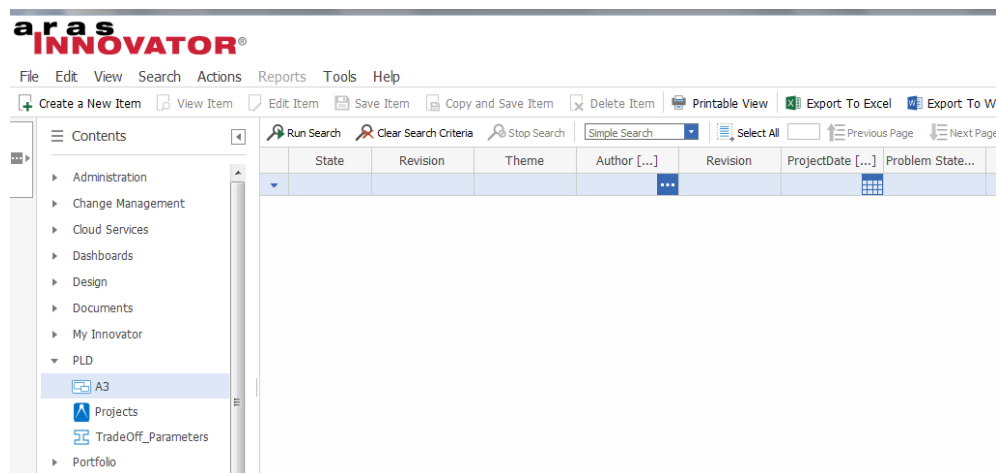


Figura 5.18: Modulo PLD

Come è possibile vedere dall'immagine precedente, l'utente (in questo caso l'amministratore) avrà la possibilità di creare un nuovo progetto, o direttamente un nuovo documento A3. Per il momento non si spiegherà il motivo dell'esistenza dell'item type "TradeOff_Parameters", sarà più chiaro tornandoci in seguito.

Anche per l'A3 e per l'item type "Projects" è possibile visualizzare tutti gli item presenti oppure crearne di nuovi mediante il pulsante "Create a new Item": tutte queste funzionalità sono messe a disposizione del programma.

5.1.3 Implementazione delle metodologie

5.1.3.1 Projects

Nei primi capitoli della tesi si è parlato dell'importanza di individuare sin dalle prime fasi della progettazione, i rischi cui potrebbe essere soggetto il progetto stesso.

La definizione dell'item type "Projects" è stato creato proprio per riuscire a rispondere a questa esigenza.

http://localhost - Project - 205DF86EFBF34FF5B2C8B4591E1EC774 - Mozilla Firefox

File Edit Views Search Actions Reports Tools Help

Create a New Item Save Item Delete Item Refresh Item Printable View Export To Excel

Project

Name: Project1 Revision:

Author: Michael Swenty Date: 5/2/2014

Description: Woodchuck Custom Closets

Created By: Innovator Admin
Created On: 5/2/2014
Modified By: Innovator Admin
Modified On: 5/2/2014
Locked By: Innovator Admin
Major Rev: A
Generation: 1
State:

Project Risk Project Team

Actions Create Related New relationship Pick/Replace Related Item Delete relationship

| Description | Probability Occ... | Impact | Ranking |
|-------------|--------------------|--------|---------|
|-------------|--------------------|--------|---------|

Ready 0 Items found. Aras Innovator

Figura 5.19: Finestra di dialogo per la creazione di un nuovo progetto

Com'è possibile vedere dalla figura, la maschera che è mostrata all'utente quando decide di creare un nuovo progetto, è divisa in due sezioni differenti: nella prima, nella parte superiore della finestra, sono presenti tutte le informazioni riguardanti il progetto, come il nome, il nome della persona responsabile e la data in cui è stato creato. Oltre a queste, sono

presenti delle informazioni, non modificabili, sulla revisione, che sono gestite in maniera automatica, e trasparente all'utente, dal programma.

Nella parte inferiore della finestra invece si hanno le così dette "Relationships": in questo caso avremmo due diversi tipi di relationships, una per la gestione del team associato al progetto, e per la gestione e visualizzazione dei rischi.

Nei capitoli precedenti si è descritta in dettaglio la metodologia che sarà utilizzata per la gestione di tali rischi: nell'immagine è possibile vedere che sono presenti i due fattori che s'intendono considerare per ogni rischio, impact e probability of occurrence, e il ranking, il cui valore sarà la moltiplicazione del valore associato ai due parametri precedenti.

La relationship "Project Team" permette di inserire diverse figure professionali all'interno di un team per la gestione del progetto. Data la sua semplicità, non ci soffermeremo ulteriormente su quest'aspetto.

Maggiore attenzione merita invece la gestione dei rischi. Nella figura sottostante è possibile vedere la finestra che appare nel momento in cui si decide di aggiungere un nuovo rischio: si nota come non sia possibile modificare l'elemento con etichetta "Create Related". Questo indica che la relazione che si andrà a creare è di tipo dipendente. Questi tipi di relazioni stanno a indicare che è possibile creare un nuovo item (in questo caso l'item che gestisce un nuovo rischio) solamente nel contesto di una relazione. Ciò significa che nel nostro caso non sarà possibile creare un rischio scorporato dalla relazione.

Questa caratteristica non è da confondere con le "null relationships". Queste sono delle particolari forme di relazioni in cui non è presente un valore per la proprietà "related_id", e tutte le informazioni che s'intendono salvare, saranno gestite mediante proprietà della relazione stessa.

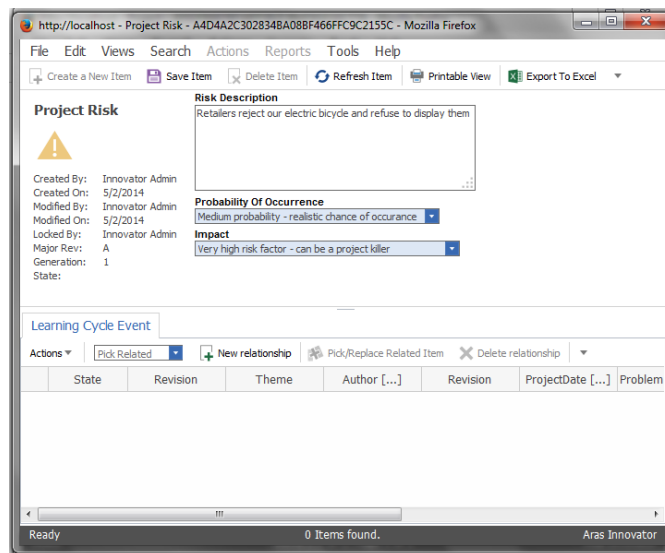


Figura 5.20: Finestra di dialogo per la gestione dei rischi

Anche in questo caso possiamo notare come la finestra sia divisa in sezioni differenti. Nella sezione in alto si ha la possibilità di inserire una descrizione per il rischio che s'intende gestire e di selezionare un diverso valore per i due parametri "probability of occurrence" e "Impact": tramite i menù a tendina è possibile scegliere tra cinque differenti valori pre impostati. In questo caso non sarà possibile salvare direttamente da questa finestra (si può notare come il pulsante per il salvataggio sia disabilitato), ma si dovrà chiudere la finestra e salvare direttamente l'intero progetto (i rischi al progetto non esistono come entità a se stanti, ma solamente come relazione con un progetto).

Project Risk

Project Team

Actions

Create Related

New relationship

Pick/Replace Related Item

Delete relationship

Export To Excel

Export To Word

| | Description | Probability Occurrence | Impact | Ranking |
|--|----------------------|---|---|---------|
| | Retailers reject ... | Medium probability - realistic chance of occurrence | Very high risk factor - can be a project killer | 15 |

Figura 5.21: Salvataggio nuovo rischio

Una volta che il rischio è stato creato per quel particolare progetto, sarà calcolato automaticamente il valore del ranking. Questo valore sarà molto utile quando si avranno diversi rischi da dover gestire per ogni singolo progetto: per capire quale affrontare prima basterà prendere quello con il valore di questo parametro più elevato.

Nella figura Figura 5.20: Finestra di dialogo per la gestione dei rischi non si è descritto il significato della seconda sezione. Come si può vedere esiste una relazione per l'item type "Project Risks" denominata "Learning Cycle Event". Come si è avuto modo di spiegare in precedenza il "Learning cycle event" non è altro che un mini progetto cui è assegnato un team, e che è eseguito per la mitigazione di un rischio.

La gestione di questo mini progetto è fatta attraverso l'item type "A3".

5.1.3.2 A3

Se si da uno sguardo alla figura Figura 5.18: Modulo PLD si nota com'è possibile visualizzare tutti i documenti A3, mediante la selezione della voce A3 dal menu PLD, che sono stati creati. Oltre alla visualizzazione è possibile creare dei nuovi documenti. Questa scelta potrebbe stonare con quella fatta in precedenza per la gestione dei rischi: com'è stato sottolineato, in quella situazione, non è possibile creare un rischio se non associandolo direttamente a un progetto.

La scelta di lasciare la possibilità di creare dei documenti A3, senza associarli direttamente a un "Learning Cycle", è stata fatta poiché la gestione di questo tipo di attività non è la sola per la quale si usa un documento A3: si potrebbe avere la possibilità di gestire un progetto, una modifica, o qual si voglia attività mediante questo strumento.

Per linearità la spiegazione dell'item type "A3" continuerà sulla falsa riga di quella utilizzata per la spiegazione degli item type precedenti.

Una volta creato un nuovo rischio, è possibile associarli un documento A3 creando una nuova relazione di tipo "Learning Cycle Event". In questo caso sarà possibile scegliere tra

un A3 esistente, e quindi creato in precedenza, oppure creare uno nuovo. Per la nostra spiegazione utilizzeremo quest'ultima possibilità.

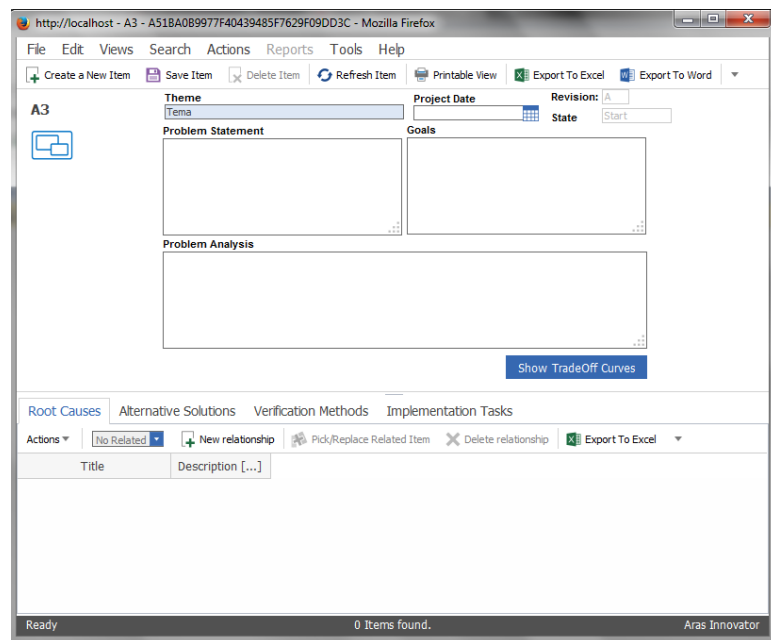


Figura 5.22: Creazione di un nuovo documento A3

La finestra che è mostrata all'utente è più ricca rispetto alle precedenti. Anche in questo caso avremmo la classica suddivisione in due sezioni: qui il numero di relazione è decisamente maggiore.

Nella parte altra della finestra avremo un campo "Theme" per la specifica del tema, o del soggetto, dell'A3; un campo "Project Data" per specificare la data di creazione o di modifica del documento; delle text area per indicare, rispettivamente, i problemi che s'intendono affrontare (Problem Statement), gli obiettivi (Goals) e l'analisi del problema (Problem Analysis). Anche in questo caso sarà presente un campo per la gestione delle revisioni, più un campo per la gestione dello stato.

La spiegazione delle relazioni parte con la descrizione delle "Root Causes". Anche in questo caso siamo in presenza di una relazione nulla: sarà possibile indicare un nome per la causa radice che si è trovata, insieme con una descrizione.

La relazione “Alternative Solutions” permette di creare o inserire diverse possibili soluzioni al problema in analisi. In seguito si andrà ad analizzare la creazione di una nuova soluzione.

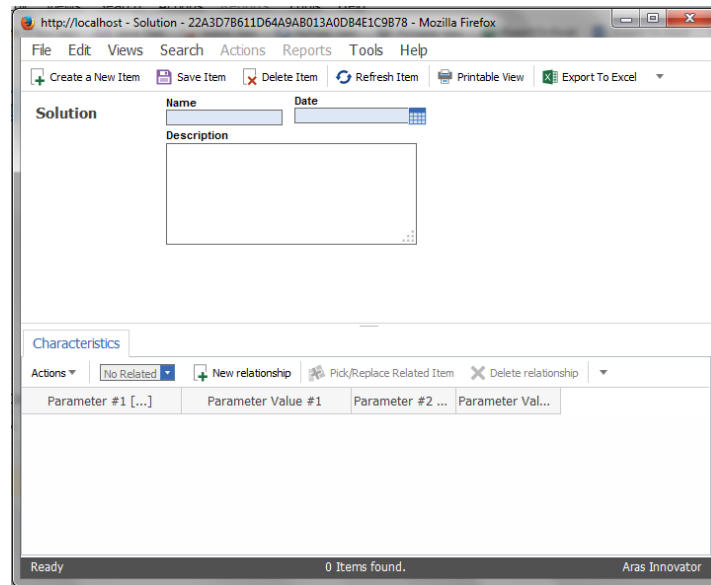


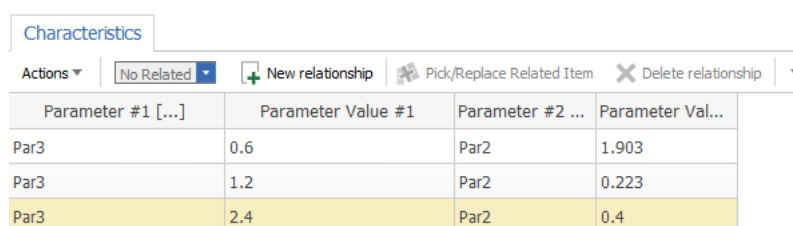
Figura 5.23: Creazione di una nuova soluzione

Anche in questo caso sarà mostrata la tipica interfaccia per la creazione di un nuovo item. Dopo avere inserito un nome, una data e una descrizione per la soluzione si possono inserire quelle che sono state chiamate caratteristiche (“Characteristics”) della soluzione.

Con il termine “caratteristica” si fa riferimento ai diversi parametri che è possibile analizzare per una data soluzione o per un dato problema. Per esempio, se si sta analizzando il motivo per il quale la cinghia di trasmissione di un veicolo si rompe, i parametri da considerare potrebbero essere il numero di chilometri percorsi e la forza esercitata dal motore sulla cinghia stessa.

La scelta di questi parametri deve essere fatta con particolare attenzione, in quanto una loro scelta errata potrebbe compromettere lo studio o non far guadagnare informazioni aggiuntive al team di progettazione. Uno studio di queste tematiche esula dagli scopi della tesi: lo strumento che stiamo analizzando è posto a valle di queste analisi.

Dalla figura si può notare che la relazione ha quattro diversi valori, tutti creati come proprietà della relazione stessa: i primi due parametri indicano il nome e il valore del primo parametro, mentre il terzo e quarto parametro indicheranno nome e valore del secondo parametro. Nella figura sotto stante è possibile vedere la situazione dopo l'inserimento di tre coppie di valori. È da notare che i parametri da utilizzare per lo studio devono essere tutti uguali (ciò vuol dire che se si sceglie di utilizzare Par3 come primo parametro, bisognerà continuare a utilizzare lo stesso parametro; non avrebbe senso usare parametri diversi).



| Parameter #1 [...] | Parameter Value #1 | Parameter #2 ... | Parameter Val... |
|--------------------|--------------------|------------------|------------------|
| Par3 | 0.6 | Par2 | 1.903 |
| Par3 | 1.2 | Par2 | 0.223 |
| Par3 | 2.4 | Par2 | 0.4 |

Figura 5.24: Inserimento valori per i parametri della soluzione

Precedentemente è stata rimandata la spiegazione di un valore del modulo PLD, e cioè “TradeOff_Parameters”. Quando si va a inserire la coppia di valori per la caratteristica associata a ogni soluzione, non si fa altro che selezionare un item di tipo “TradeOff_Parameters” e associarli un valore numerico. Quello che si è tralasciato di specificare è come sono creati questi parametri, in quanto mediante la relazione “Characteristics” di ogni soluzione è possibile solo selezionare dei valori già esistenti.

Mediante la voce del TOC “TradeOff_Parameters” è possibile visualizzare e creare quanti più parametri si vogliono, e associare a ognuno di essi una descrizione. Chiaramente tale possibilità è riservata solo per l'amministratore o per il proprietario (il responsabile) del progetto.

Come abbiamo avuto modo di sottolineare la scelta dei parametri è un'attività molto delicata. La scelta è dunque completamente slegata dalla definizione e gestione del documento A3.

Continuando con la spiegazione delle relazioni presenti nell'item type A3, si può passare alla relazione “Verification Methods”. Quando si è spiegato l'utilizzo e la costruzione di un documento A3, si è sottolineato come, assieme alla definizione di un numero diverso di soluzioni, sia importantissimo definire una serie di metodologie per la verifica delle soluzioni trovate. La relazione quindi permette, per ogni soluzione inserita nella relazione precedente, di individuare un metodo di verifica. La metodologia in questo caso sarà data mediante una descrizione testuale: si potranno inserire i test o le simulazioni effettuate.

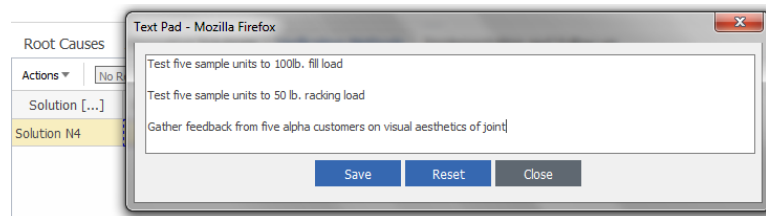


Figura 5.25: Finestra di dialogo per l'inserimento dei metodi di valutazione per una soluzione

L'ultima relazione da analizzare è quella che è stata chiamata “Implementation Tasks”. Mediante questa relazione è possibile costruire un diagramma temporale per la gestione delle attività d'implementazione della soluzione trovata.

Sarà possibile inserire il nome dell'attività da compiere, la data d'inizio attività, le settimane necessarie e un responsabile.

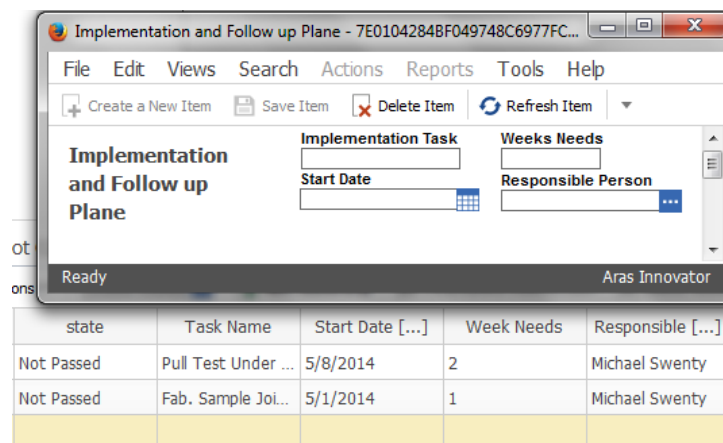


Figura 5.26: Creazione del piano

Per ogni attività (task) che viene creata, è possibile specificare se è stata portata a termine oppure meno: la gestione di questa possibilità è fatta tramite un oggetto che in ARAS è chiamato “life cycle map”.

Il life cycle map non è altro che una serie di stati tra i quali un item può transitare durante la sua esistenza. In un life cycle è quindi costituito da stati e transizioni, che altro non sono che cammini tra i diversi stati. Una volta che è stato creato un life cycle, ed è stato associato a uno specifico item type, è possibile specificare quale sarà lo stato preliminare, o meglio lo stato in cui l’item si troverà appena creato. Chiaramente durante la vita dell’item ci sarà necessità di farlo transitare da uno stato all’altro; questo è reso possibile attraverso l’uso del comando “promote” disponibile per tutti gli item.

L’utilizzazione di questo strumento permette di creare mappe, anche molto complesse: la creazione avverrà attraverso l’utilizzazione di un’interfaccia grafica molto semplice, attraverso la quale sarà possibile aggiungere i diversi stati, aggiungere le transizioni, e specificare una serie di parametri (i permessi necessari per la promozione di un item, i metodi da attivare prima e dopo la promozione e così via) sia per gli stati, sia per le transizioni. Nella figura seguente si mostra il life cycle utilizzato per la gestione dell’item type “Implementation Tasks”.

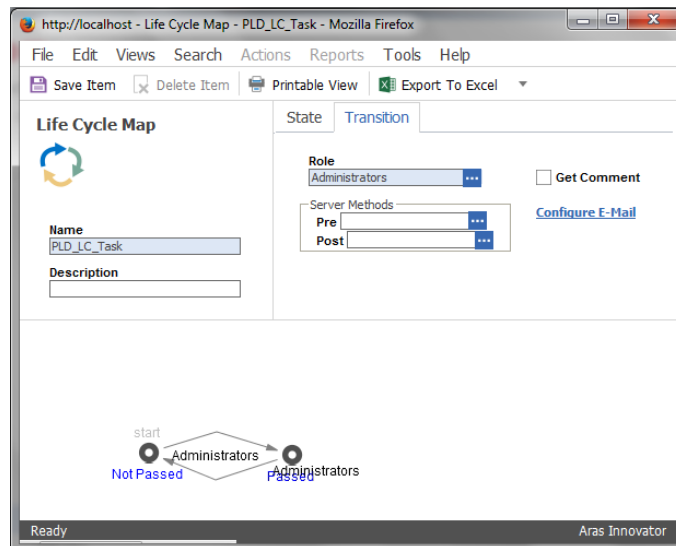


Figura 5.27: Gestione del ciclo di vita per un attività del piano implementativo

Come si può vedere la mappa che è stata creata, è molto semplice: sono presenti solamente due stati, il primo, “not passed”, è lo stato iniziale.

I due stati indicheranno, rispettivamente, che l’azione o task del piano è stata eseguita oppure non è stata eseguita. Come si è avuto modo di specificare precedentemente, la promozione avviene mediante un comando predefinito:

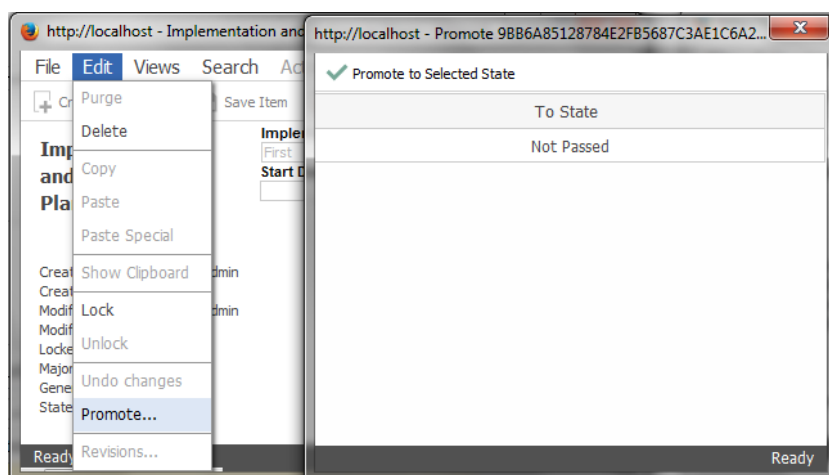


Figura 5.28: Promozione di un attività

Nella figura l'item si troverà nello stato "passed", e lo si vuole far transitare verso lo stato "not passed".

È da sottolineare come la transizione da uno stato all'altro sia resa possibile dal fatto che nella mappa sono presenti i cammini da uno stato all'altro: per esempio se non fosse stato inserito il cammino da "passed" a "not passed", una volta che l'item è stato dichiarato "passed", non sarebbe possibile tornare indietro (nel nostro caso è stato aggiunto poiché è sempre possibile che si verifichino errori, e che sia necessario far la transizione da "passed" a "not passed").

5.1.3.3 Trade-Off curves

Se si dà uno sguardo alla Figura 5.22: Creazione di un nuovo documento A3 si può notare che è presente un pulsante per la visualizzazione delle trade-off curves.

Come si è avuto modo di spiegare in dettaglio nei capitoli precedenti, le trade-off curves sono uno strumento molto potente per la visualizzazione e per il confronto delle performance di soluzioni alternative.

Una volta che sono state specificate le soluzioni per il nostro A3, nella sezione etichettata "Alternative Solutions", e una volta inserite i parametri con i rispettivi valori, è possibile creare e visualizzare le curve.

Per la visualizzazione delle curve con ci si è serviti degli strumenti del programma ARAS, ma si è creata una pagina HTML esterna che legga i dati salvati nell'A3 elaborandoli per creare le curve. Nella finestra successiva è possibile visualizzare l'output di questa elaborazione.

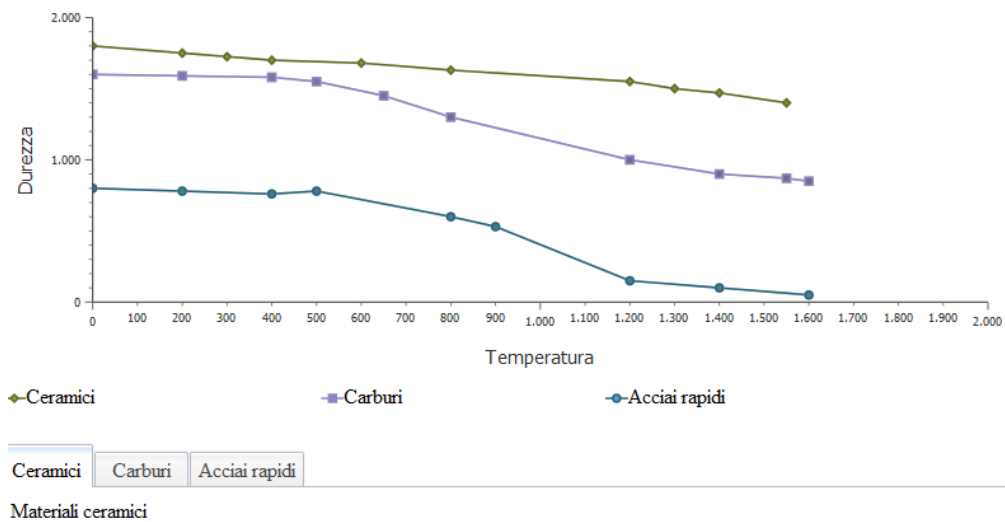


Figura 5.29: Trade-off curves

Com'è possibile notare, è presente una sola finestra grafica per tre diverse curve, e quindi tre diverse soluzioni proposte per il problema. Sul grafico sono riportati i nomi dei parametri considerati per la soluzione, e la leggenda per distinguere le performance delle diverse soluzioni.

Sotto la finestra grafica è presente una seconda sezione con un menu a etichette. In ognuna delle etichette sarà presente una breve descrizione della soluzione considerata: anche in questo caso i dati della descrizione saranno prelevati direttamente dall'A3.

Di seguito due ulteriori esempi sulle curve di trade-off.

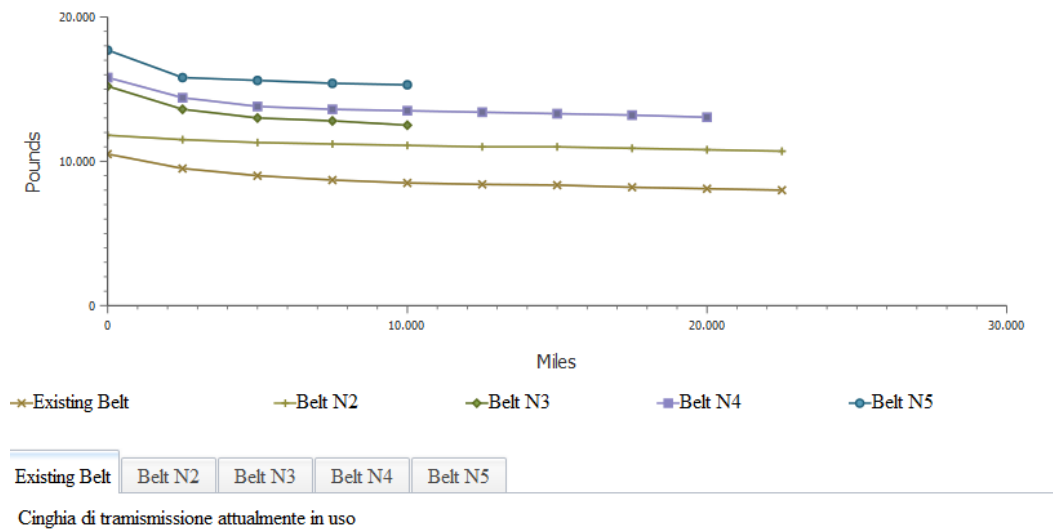


Figura 5.30: Trade-off curves caso Harley Davidson

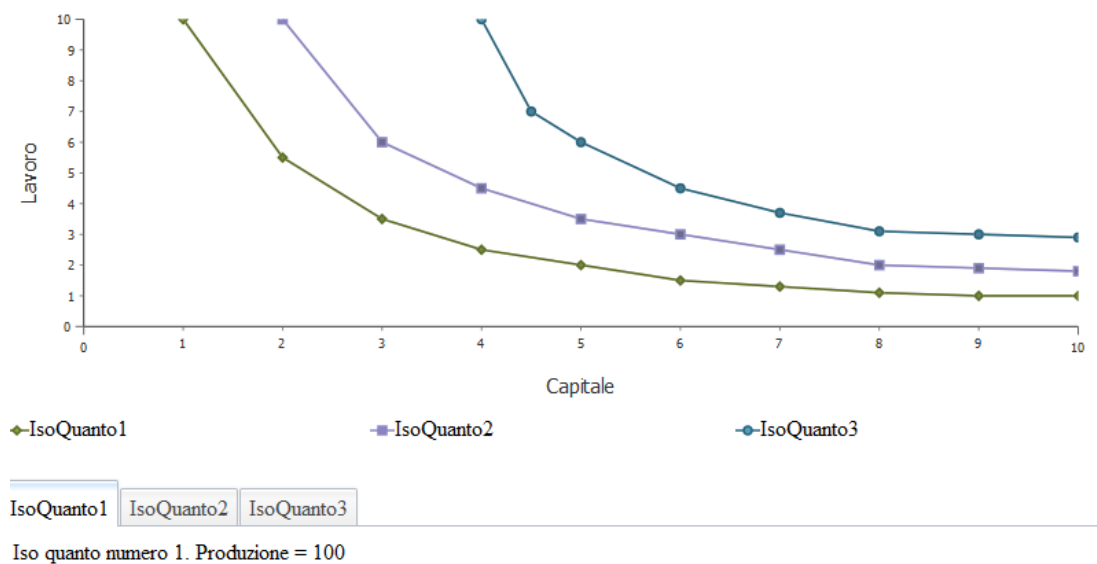


Figura 5.31: Trade-off curves caso Produzione

Essendo il software sostanzialmente un'applicazione web, costruita tramite pagine HTML e Javascript, è stato possibile ricavare i valori dell'A3 semplicemente utilizzando le funzioni base di Javascript insieme alle funzioni messe a disposizione dall'Innovator Object Model (IOM). Lo IOM è una collezione di classi e metodi che possono generare,

inviare e rispondere a richieste fatte sul server. Queste funzionalità possono essere accedute sia dalla parte server (attraverso il linguaggio C# o Visual Basic), sia dalla parte client (attraverso il linguaggio Javascript). Gli scopi principali di queste funzioni sono di mettere a disposizione un'interfaccia unica per la gestione degli item come entità astratte.

5.1.4 Gli Item Types Utilizzati

5.1.4.1 Introduzione

Per meglio comprendere la logica dietro al modulo che è stato appena descritto, è utile dare uno sguardo d'insieme agli item types che sono stati generati, come si legano tra di loro (quali sono le loro relazioni) e quali proprietà sono state definite per la loro gestione. Per semplicità si farà riferimento solamente alle proprietà custom, e cioè quelle definite per gli scopi del modulo, tralasciando le proprietà di default di ogni item type.

Nella figura seguente è possibile vedere come ogni entità si lega alle altre.

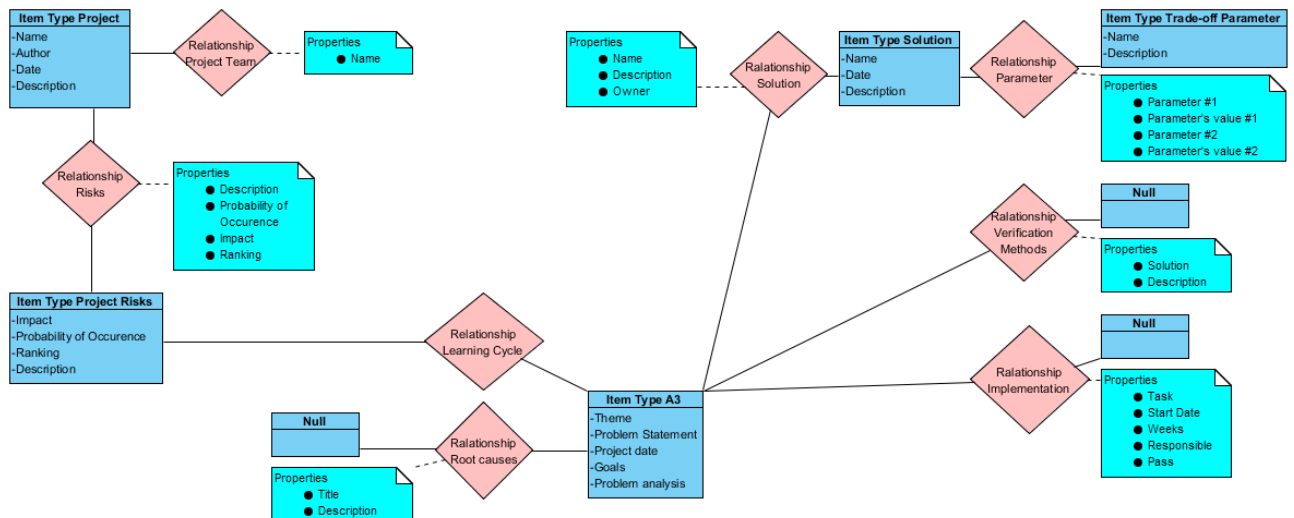


Figura 5.32: Mappa degli item e delle relazioni usate

■ Indica un Item Type e le sue proprietà

■ Indica una relazione tra due Item Type (se uno dei due è null, si è in presenza di una null relationship)

■ Indica le proprietà della relazione

5.1.4.2 Item type Project

L'item type denominato *Project* è l'entità principale del modulo: attraverso esso, infatti, è possibile creare un nuovo progetto e definire i suoi rischi (se presenti).

Oltre alle proprietà standard, sono state aggiunte altre quattro proprietà:

- Name: è il nome che è attribuito al progetto; mediante esso è possibile fare delle ricerche sul database, è quindi opportuno utilizzare dei nomi esplicativi.
- Author: è il nome dell'autore, o del responsabile, del progetto; la scelta sarà fatta andando a selezionare un utente salvato all'interno del programma (quest'utente avrà una propria username e password per accedere al programma).

- Date: la data di creazione del progetto.
- Description: descrizione sintetica dello scopo del progetto.

Com'è stato detto in precedenza, la modifica di queste proprietà è fatta, da parte dell'utilizzatore, attraverso una form associata all'item type: tale form è completamente personalizzabile.

5.1.4.3 Item Type Project Risk

L'item type *Project risk* è l'entità demandata alla gestione del rischio associato a uno specifico progetto.

Le proprietà che sono state definite sono le seguenti:

- Impact: valore numerico su una scala da 1 a 5.
- Probability of occurrence: anche in questo caso si tratterà di un valore numerico da 1 a 5.
- Ranking: valore numerico calcolato come $\text{Impact} * \text{Probability of Occurrence}$.
- Description: descrizione sintetica del rischio.

Anche in questo caso la creazione, o la modifica di un rischio avviene con una forma personalizzata.

5.1.4.4 Item Type A3

L'item type A3 permette di definire un nuovo “documento”A3. Com'è stato possibile osservare nella descrizione dell'interfaccia, saranno presenti diverse proprietà personalizzate:

- Theme: permette di specificare il tema, o lo scopo dello studio condotto mediante lo strumento A3.
- Problem statement: la proprietà è utilizzata per fornire una breve descrizione del problema oggetto di studio.
- Project date: la data di creazione.
- Goals: indica gli obiettivi dello studio.
- Problem Analysis: una volta definiti e descritti i problemi oggetto di studio, si può passare alla loro analisi.

Oltre a queste si è fatto uso della proprietà per la gestione delle revisioni dei documenti.

5.1.4.5 Item TypeSolution

L'entità *Solution* permette di istanziare una nuova soluzione per un determinato problema.

È stato opportuno definire le seguenti proprietà aggiuntive:

- Name: nome della soluzione.
- Date: data di creazione.
- Description: breve descrizione della soluzione.

5.1.4.6 Item Type Trade-off Parameters

La gestione dei parametri delle curve di Trade-off è fatta in maniera molto semplice.

È stato necessario aggiungere solamente due parametri:

- Name: il nome associato al parametro; dovrebbe essere un nome abbastanza esplicativo.
- Description: descrizione del parametro.

5.1.5 Relationship

Come è stato detto precedentemente, oltre alla definizione degli item type, è stato necessario definire le così dette *Relationship* (è da notare che per il programma le relationships sono item type). Successivamente sarà data una breve descrizione per ognuna di esse, con l'indicazione delle proprietà utilizzate. Prima di procedere è utile ricordare che in molti casi è stata utilizzata la possibilità di creare le *null relationships*. Questi particolari tipi di relazioni hanno un source item ma non hanno un related item.

5.1.5.1 Relationship Risks

Questa relazione esiste tra gli item type *Project* e *Project Risk*. Permette di definire, uno o più rischi associati a un progetto.

Per la sua gestione non è stato necessario definire alcuna proprietà aggiuntiva.

5.1.5.2 Relationship Root Causes

Questa relazione è il primo caso di null relationship. È stato deciso di preferire questo tipo di soluzione perché non si aveva la necessità di definire un item type specifico per le cause.

La loro gestione è stata dunque implementata utilizzando le proprietà della relazione.

Sono state, dunque, definite le seguenti proprietà:

- Title: sostanzialmente è il nome associato alla causa in esame.
- Description: breve descrizione della causa.

5.1.5.3 Relationship Solution

Anche in questo caso è stata utilizzata una null relationship.

Sono state definite le seguenti proprietà:

- Name: questa proprietà è di tipo Solution (Item type Solution); sarà quindi possibile creare una nuova soluzione, oppure selezionare una soluzione già esistente.
- Description: breve descrizione della soluzione prescelta.
- Owner: proprietario (cioè chi ha creato la soluzione) della soluzione; anche in questo caso la scelta del proprietario sarà fatta tra gli utenti del software.

5.1.5.4 Relationship Verification Methods

La relazione permette di definire, per ognuna delle soluzioni indicate nella precedente relazione, un metodo di verifica. Il metodo sarà descritto in maniera testuale mediante le seguenti proprietà:

- Solution: proprietà di tipo Solution; la scelta permessa all'utente sarà confinata tra le soluzioni specificate per quell'A3.
- Description: descrizione del metodo utilizzato per la verifica della soluzione.

5.1.5.5 Relationship Implementation

Questa relazione permette di definire un piano implementativo per la soluzione trovata.

Le proprietà definite sono:

- Task name: indicazione del nome dell'attività da svolgere.
- Start date: la data d'inizio dell'esecuzione dell'attività specificata.
- Weeks: le settimane necessarie a completare l'attività: il tipo della proprietà è numerico.
- Responsible: si indica l'utente responsabile del completamento dell'attività

5.1.5.6 Relationship Parameters

Quest'ultima relazione permette di definire i parametri che saranno successivamente letti per la costruzione delle curve di trade-off. Essendo utilizzati per la costruzione delle curve, i parametri saranno inseriti in coppie (un'ascissa e un'ordinata).

Le proprietà definite sono:

- Parameter #1: il nome del primo parametro (come nel caso della relazione Solution, anche qui la proprietà sarà di tipo Trade-off Parameter).
- Parameter's value #1: il valore associato al primo parametro.
- Parameter #2: il nome del secondo parametro
- Parameter's value #2: il valore associato al secondo parametro.

5.1.6 I metodi

In questa sezione s'illustreranno i metodi utilizzati e le loro caratteristiche.

I metodi in ARAS sono suddivisi in metodi lato client e metodi lato server. È comunque lasciata la possibilità nei metodi lato client, di invocare metodi lato server. I due tipi di metodi utilizzano un linguaggio di programmazione diverso: lato client si utilizzerà il linguaggio javascript, mentre lato server si avrà la possibilità di scrivere il programma o attraverso il linguaggio visual basic, o attraverso il linguaggio C#.

Come si è detto il programma mette a disposizione del programmatore lato client tutte le funzioni con cui è possibile interagire con il database e con il server, attraverso l'IOM

(Innovator Object Model): l'IOM è una collezione di classi e metodi che generano, inviano e rispondono a richieste fatte sul server.

5.1.6.1 Metodi lato Client

5.1.6.1.1 PLD_Get_Solution_Name

Questo, così come tutti gli altri metodi lato client, metodo è stato inserito in un file javascript, denominato innovator.js, che sarà utilizzato nel momento in cui l'utente decide di visualizzare le curve di trade-off per uno specifico A3.

Come si può intuire dal nome, il metodo restituisce tutte le soluzioni per uno specifico A3, il cui ID sarà specificato come parametro d'ingresso.

Per riuscire ad ottenere il risultato, si vanno a recuperare tutte le relazioni di tipo "Alternative Solution", e per ognuna di esse si recupera il nome della soluzione specificata. Chiaramente la relazione conterrà le informazioni essenziali all'interno dei parametri "source id", dove sarà contenuto l'id dell'A3, e "related id", dove sarà contenuto l'id della soluzione. Per riuscire a recuperare il nome della soluzione sarà necessario recuperare il suo id, e tramite questo fare una ricerca tra tutte le soluzioni per identificare il suo nome.

Il risultato sarà dunque un array, in cui per ogni cella sarà presente un nome.

5.1.6.1.2 PLD_Get_Solution_Description

Questo metodo è molto simile al precedente. Come parametro d'ingresso avremmo il nome della soluzione: uno dei vincoli di progettazione è stato quello di avere dei nomi per le soluzioni univoci. Questo permette, attraverso il nome, di eseguire delle ricerche tra tutte le soluzioni presenti nel database, e quindi recuperare facilmente la descrizione.

Il valore restituito dalla funzione sarà la stringa contenente la descrizione.

5.1.6.1.3 PLD_Get_Parameter

Questa funzione permette di recuperare i parametri definite per una data soluzione. Come parametri d'ingresso avremmo: l'id dell'A3 e il nome della soluzione.

Per riuscire a capire la logica dietro la costruzione di questo metodo, ricordiamo che tra ogni item type "Solution" e item type "TradeOff Parameter" esiste una relazione che li unisce.

Per riuscire a recuperare la soluzione corretta si utilizzerà il nome specificato come parametro e l'id dell'A3, anch'esso specificato come parametro: la specifica dell'id dell'A3 è indispensabile in quanto è possibile selezionare la medesima soluzione per A3 differenti, ognuna con i propri valori per i parametri (questi, infatti, sono salvati nella relazione tra "Solution" e "TradeOff Parameter").

Una volta identificata la relazione per la soluzione corretta, è possibile andare a leggere il nome dei parametri con i relativi valori, e salvarli in un array multidimensionale. Tale array sarà restituito in output.

5.1.6.1.4 Draw_Chart

Questo metodo è stato definito su un file separato rispetto ai precedenti metodi. Questa scelta è stata fatta per tenere separate le funzioni per il recupero dei dati salvati sul PLM, da quelle per la visualizzazione grafica degli stessi dati.

I parametri d'ingresso sono i nomi delle soluzioni (quindi l'array che li contiene) e l'id dell'A3.

Utilizzando le funzioni definite in precedenza, è possibile recuperare in maniera molto semplice tutte le informazioni necessarie: in questo caso dato il nome della soluzione e l'A3 sarà necessario solamente recuperare i parametri definiti per la stessa soluzione. Una volta recuperati questi parametri, e formattati in maniera tale da essere riconosciuti dalla libreria javascript DOJO, è possibile invocare le funzioni per disegnare il grafico.

La scelta di utilizzare la libreria DOJO è stata fatta in quanto lo stesso programma ARAS utilizza tale libreria ma nulla vietava di utilizzare un'altra libreria javascript.

Le specifiche per il disegno del grafico impongono il settaggio di diversi parametri tra cui: i due assi (indicando il nome, e i parametri sulla lunghezza), lo stile grafico, il tipo di grafico (sono presenti diverse alternative che permettono di creare grafici a torta, a colonne, a linee e così via) ecc.

Oltre a queste caratteristiche è stata aggiunta una leggenda per riuscire a capire a quale linea del grafico corrisponde ciascuna soluzione trovata.

5.1.6.1.5 PLD_Custom_Solution_Search

Questa funzione è stata creata per personalizzare la ricerca tra gli item di tipo “Solution”: questa necessità sorge nel momento in cui si deve andare a selezionare un metodo di verifica per ogni soluzione, all’interno dell’A3.

Nella relazione “Verification Methods” è, infatti, necessario selezionare una soluzione: se non ci fosse questa personalizzazione, quando l’utente ricerca tra le soluzioni potrebbe, erroneamente, selezionare una soluzione che non è stata definita per il suo A3. Proprio per questo motivo si è deciso di implementare un metodo che rendesse possibile all’utente ricercare solo tra le soluzioni che lui, o chi per lui, ha definito nella sezione “Alternative Solutions” dell’A3.

Per riuscire nell’intento, ARAS, mette a disposizione dei metodi specifici per andare a modificare il risultato della query che popolerà la griglia di ricerca. Tale modifica sarà effettuata andando a invocare il metodo “inArgs.QryItem.item.setAttribute(idlist)” e passandogli come parametro una stringa con tutti gli id delle soluzioni che s’intendono restituire all’utente (gli id dovranno essere separati da una virgola).

Conclusioni

Il progetto svolto nell'ambito della tesi aveva come scopo principale quello di estendere le funzionalità messe a disposizione dal software ARAS Innovator tramite la realizzazione di un nuovo modulo. L'obiettivo di questo modulo è di fornire dei nuovi strumenti a supporto della progettazione in maniera tale da riuscire a cogliere i vantaggi delle metodologie della concurrent engineering e del lean development.

Attraverso questo modulo le aziende possono incominciare ad utilizzare gli strumenti A3 e Trade-off curves, e accorgersi dell'effettivo valore aggiuntivo nella progettazione e nello sviluppo dei nuovi prodotti, e al tempo stesso riuscire ad avere un software in grado di integrare questa nuova gestione con i tradizionali strumenti di gestione del ciclo di vita del prodotto, tipici di un software PLM.

Durante le mie ricerche ho potuto notare un disinteresse alle problematiche del cliente, e soprattutto un marcato scetticismo verso nuove metodologie di progettazione, quali possono essere quelle lean e quelle introdotte con la set-based concurrent engineering. Queste metodologie sono state applicate da numerose aziende, anche di grossa dimensione, con grossi benefici nella qualità del lavoro e nella riduzione dei costi di progettazione e produzione. Questa, purtroppo, è in contro tendenza con la situazione delle PMI italiane.

Purtroppo non è stato possibile testare il modulo sviluppato in una situazione reale all'interno di un'azienda, ma già dai casi provati si è potuto notare un'efficace gestione della progettazione e dei problemi ad essa collegati.

Ringraziamenti

Finalmente ho raggiunto il mio sogno. Ha raggiunto il traguardo per cui sono venuto qua in Toscana qualche anno fa. È stato un cammino lungo e faticoso, ma al tempo stesso gratificante. Ora sono un Ingegnere.

In quest'ultima pagina vorrei fare qualche doveroso ringraziamento.

Innanzitutto ringrazio il professor Chiavaccini e suo figlio Guido per il costante supporto durante il mio lavoro di tesi, e per avermi fatto conoscere nuove tematiche e nuove possibilità, anche in vista della mia carriera professionale: sono stati mesi di lavoro molto interessanti e stimolanti.

Un ringraziamento speciale va ai miei genitori per il loro appoggio e per l'amore che ogni giorno mi dimostrano. Senza i vostri sacrifici oggi non sarei quello che sono e non sarei qui a scrivere questi ringraziamenti e a festeggiare questo giorno. È a voi che voglio dedicare questo lavoro, è a voi che dedico la mia laurea.

Grazie ai miei fratelli, siete la mia ispirazione.

Grazie a Giulia. Quattro anni fa hai scelto di seguirmi. Non è stato sempre semplice, ma abbiamo resistito. È anche a te che voglio dedicare questo giorno, perché so i sacrifici che hai fatto per starmi vicino. Grazie per avermi sopportato e incoraggiato durante i momenti negativi, devo tanto anche a te. Ora un nuovo inizio, insieme.

Grazie ai nuovi e vecchi amici.

Appendice

Lista delle figure

| | |
|---|-----|
| Figura 1.1: Waterfall model | 9 |
| Figura 1.2: Tradizionale approccio Point-Based allo sviluppo prodotto..... | 12 |
| Figura 1.3: Traditional vs. Set-Based Concurrent Engineering | 21 |
| Figura 1.4: Differenze di approccio per la mitigazione dei rischi | 27 |
| Figura 1.5: Il sistema 1-to-5 | 29 |
| Figura 1.6: Un esempio di rischi identificati per il progetto di una bici elettrica | 30 |
| Figura 1.7: Rappresentazione grafica del sistema 1-to-5 | 30 |
| Figura 1.8: Un gap d'informazione esiste quando c'è disparità tra l'informazione corrente di un team e le informazioni necessarie per ottenere un progetto con bassi rischi | 32 |
| Figura 1.9: Il numero di iterazione dipende dalla complessità del problema che si sta affrontando | 33 |
| Figura 1.10: Un tipico documento che sintetizza l'agenda di un learning cycle | 34 |
| Figura 2.11: L'A3 è un metodo strutturato per l'implementazione del ciclo Plan / Do / Check / Act, per il miglioramento continuo | 37 |
| Figura 2.12: Il classico diagramma a spina di pesce, per la decomposizione di un problema nelle sue potenziali cause radici | 45 |
| Figura 4.13: Stack Aras Innovator..... | 84 |
| Figura 4.14: Architettura Aras Innovator | 85 |
| Figura 4.15: Comunicazione tra client e server mediante messaggi AML | 88 |
| Figura 4.16: Relazioni tra items | 88 |
| Figura 5.17: Finestra principale di Aras Innovator | 91 |
| Figura 5.18: Modulo PLD | 92 |
| Figura 5.19: Finestra di dialogo per la creazione di un nuovo progetto | 93 |
| Figura 5.20: Finestra di dialogo per la gestione dei rischi | 95 |
| Figura 5.21: Salvataggio nuovo rischio | 95 |
| Figura 5.22: Creazione di un nuovo documento A3 | 97 |
| Figura 5.23: Creazione di una nuova soluzione..... | 98 |
| Figura 5.24: Inserimento valori per i parametri della soluzione | 99 |
| Figura 5.25: Finestra di dialogo per l'inserimento dei metodi di valutazione per una soluzione ... | 100 |
| Figura 5.26: Creazione del piano..... | 101 |
| Figura 5.27: Gestione del ciclo di vita per un attività del piano implementativo..... | 102 |
| Figura 5.28: Promozione di un attività | 102 |
| Figura 5.29: Trade-off curves..... | 104 |
| Figura 5.30: Trade-off curves caso Harley Davidson..... | 105 |
| Figura 5.31: Trade-off curves caso Produzione | 105 |
| Figura 5.32: Mappa degli item e delle relazioni usate | 107 |

Bibliografia

Riferimento a libri

Jeffrey K. Liker, The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer, McGraw-Hill, 2004.

Mascitelli R., Mastering Lean Product Development. A Practical, Event-Driven Process for Maximizing Speed, Profits, and Quality, Technology Perspectives.

Oosterwal D., The Lean Machine. How Harley-Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development, AMACON.

PMI, A guide to the Project Management Body of Knowledge, 4th Edition.

SaaKsvuori A., Immonen A., Product Lifecycle Management, 3rd Edition, Springer.

Schipper T., Swets M., Innovation Lean Development. How to Create, Implement and Maintain a Learning Culture Using Fast Learning Cycles, CRC Press.

Sobek D.K., Smalley A., Understanding A3 Thinking. A Critical Component of Toyota's PDCA Management System, CRC Press.

Stark J., Product Lifecycle Management. 21st Century Paradigm for Product Realisation, 2nd Edition, Springer.

Ward A.C., Sobek D.K., Lean Product and Process Development, 2nd Edition, 2014.

Riferimenti ad articoli

Bitran G.R., Morabito R., An Overview of Tradeoff Curves in Manufacturing Systems Design, MIT, Production and Operations Management Vol. 8, No. 1, Spring 1999.

Jacobs E.N., The characteristics of 78 Related Airfoil sections from tests in the Variable-Density Wind Tunnel, NASA.

Jensen H., Learning on Knowledge. Managing Knowledge with Lean Product Development Tools and Methods, Chalmers University of Technology, 2012.

Khan M., Set-Based Concurrent Engineering process within the LeanPPD Environment, 2011.

Ogawa S., Piller F.T., Reducing the Risks of New Product Development, MIT Sloan Management Review Vol. 47 No. 2, Winter 2006.

Olechowski A., Product development Risk Management and the Role of Transparency, MIT, 2010.

Prasad B., Sequential versus Concurrent Engineering – an Analogy, Sage Publications, 1995.

Smith P.G., Managing Risk as Product Development Schedules Shrink, Industrial Research Institute, 1999.

Sobek D.K., Ward A., Liker J.K., Toyota's Principles of Set-Based Concurrent Engineering, Sloan Management Review, Winter 1999.

Sobek D.K., Ward A.C., Principles from Toyota's Set-Based Concurrent Engineering Process, 1996.

Vassilvitskii S., Efficiently computing succinct trade-off curves, Theoretical Computer Science, 2005.

Ward A., Liker J.K., Cristiano J.J., Sobek D.K., The second Toyota Paradox: How delaying decisions Can Make Better Cars Faster, Sloan Management Review, Spring 1995.